MULTIMIX TFT: A MULTI-TASK MIXED-FREQUENCY FRAMEWORK WITH TEMPORAL FUSION TRANSFORMERS

Boje Deforce[1], **Bart Baesens**[1,2], **Jan Diels**[3], **Estefanía Serral Asensio**[1]
[1]Research Center for Information Systems Engineering, KU Leuven, Belgium
[2]Department of Decision Analytics and Risk, University of Southampton, United Kingdom
[3]Department of Soil and Water Management, KU Leuven, Belgium
{boje.deforce,bart.baesens,jan.diels,estefania.serralasensio}@kuleuven.be

ABSTRACT

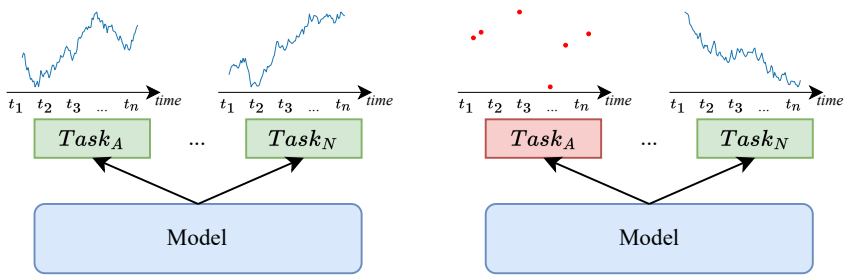
Multi-task learning (MTL) has been increasingly recognized as an effective paradigm in time-series analysis for forecasting multiple related tasks concurrently. Prior MTL frameworks for time-series forecasting have typically been devised for tasks that share the same regular time frequencies. However, numerous real-world scenarios entail tasks measured at mixed, and often irregular, time frequencies. We propose a multi-task mixed-frequency (MultiMix) learning framework for time-series forecasting that addresses the challenges of mixed-frequency scenarios where tasks are measured at different and/or irregular time intervals. Our proposed framework leverages the relationships between mixed-frequency tasks to improve accuracy and robustness of time-series forecasting across tasks. The MultiMix framework is implemented using the state-of-the-art Temporal Fusion Transformer (TFT) and is evaluated on smart irrigation, where predicting mid-day stem water potential and soil water potential pose critical challenges. The MultiMix TFT enables joint forecasting of stem water potential, measured sparsely on irregular and infrequent time intervals, and soil water potential, measured on a daily time interval. The results show substantial improvements in stem water potential prediction over state-of-the-art baselines while achieving comparable performance for soil water potential. These results confirm the effectiveness of the proposed framework for addressing the mixed-frequency time-series forecasting problem in real-world settings.

1 INTRODUCTION

Multi-task learning (MTL) is a machine learning approach that aims to improve the performance of prediction tasks by leveraging potential cross-task relationships. It has emerged as a powerful approach for solving multiple related tasks simultaneously, especially when dealing with sparsely labeled data (Zhang & Yang, 2022). By utilizing the relationships between tasks, MTL has shown its capability to improve the quality of predictions in various domains, with impressive results in computer vision (Dvornik et al., 2017), natural language processing, and speech recognition (Aghajanyan et al., 2021); see Zhang & Yang (2022) for a comprehensive survey.

Recently, the domain of time-series forecasting has made significant progress across various industries by leveraging MTL (Chen et al., 2020; Deng et al., 2022). However, existing MTL frameworks for time-series forecasting have primarily focused on addressing single-frequency problems. That is, the different tasks are restricted to a fixed and shared time interval. This limitation renders conventional MTL approaches in time-series forecasting less effective for real-world scenarios where multiple tasks are often measured at different and/or irregular time intervals (see e.g. Jiang et al. (2017) and Yang et al. (2022)). In such scenarios, one of the tasks is typically measured at fixed and regular intervals (e.g., every hour) while the other task usually contains fewer measurements captured at lower – and often irregular – intervals. We refer to such combination of tasks as mixed-frequency tasks. To overcome this challenge, we propose a multi-task mixed-frequency (MultiMix) learning framework that leverages the relationships between mixed-frequency tasks to enhance the accuracy and robustness of time-series forecasting across tasks (see Figure 1). Practically, this is achieved through a flexible MultiMix head that can be placed on top of an existing neural net architecture, allowing the model to learn from all available measurements across multiple tasks. By allowing the model to learn from all available measurements for both tasks, our framework offers an effective solution for the mixed-frequency problem. The proposed MultiMix framework presents a promising avenue for advancing the state-of-the-art in multi-task mixed-frequency time-series forecasting and has potential applications in various domains.

In this paper, we evaluate the developed MultiMix framework in the domain of smart irrigation. Smart irrigation represents a real-world mixed-frequency scenario where predicting mid-day stem water potential (ψ_{stem}) and soil



a) Conventional MTL approaches in time-series forecasting

b) Proposed MultiMix approach for mixed-frequency time-series forecasting

Figure 1: A comparison of conventional MTL approaches for time-series forecasting and our proposed MultiMix approach. In a), all tasks share the same frequency (e.g. daily) while in b), one or more tasks are captured at a different frequency than the other tasks, rendering conventional MTL approaches infeasible.

water potential (ψ_{soil}) pose critical challenges. ψ_{stem} and ψ_{soil} represent measures of the amount of water in the stem and soil, respectively. Especially ψ_{stem} is one of the best indicators on whether plants are getting enough water or are under drought stress. In turn, this information can lead to more efficient irrigation systems with freshwater savings of up to ∼30% (Martínez-Gimeno et al., 2018). While ψ_{soil} can already be measured by sensors at a high time-frequency at regular intervals, ψ_{stem} can only be reliably measured manually in specific weather conditions, and therefore typically at irregular and infrequent time intervals. As such, the available ψ_{stem}-measurements are limited. Moreover, the prediction of ψ_{stem} is non-trivial due to the influence of numerous factors such as ψ_{soil}, root system distribution, soil characteristics, atmospheric conditions, and many more (Janssens et al., 2011). This poses significant challenges in accurately forecasting ψ_{stem} using single-task models.

To solve this challenge, we apply our proposed MultiMix framework to predict both ψ_{stem} and ψ_{soil} simultaneously. We implement the MultiMix framework using the state-of-the-art Temporal Fusion Transformer (TFT) (Lim et al., 2021). The MultiMix TFT incorporates ψ_{soil}-information – measured on a daily time interval – into the prediction of ψ_{stem} – measured on irregular and infrequent time intervals – achieving a substantial improvement in the accuracy of ψ_{stem}-prediction over baseline state-of-the-art methods without compromising the performance of ψ_{soil}-prediction.

The contribution of this paper is three-fold: (i) we present the MultiMix framework; (ii) implement it with the state-of-the-art TFT; and (iii) demonstrate the empirical superiority and usefulness of the MultiMix TFT for the real-world problem of ψ_{stem}- and ψ_{soil}-prediction. Note that while we applied the model to the problem of water potential prediction, the model can be applied to any combination of tasks that are measured at mixed frequencies and/or at irregular time intervals.

2 Related Work

Substantial performance gains have been achieved by using an MTL set-up in various domains, ranging from computer vision to natural language processing, as well as cross-domain combinations of those (see Al-Rawi & Valveny (2019) and Zhang & Yang (2022) for some examples). The advantages of MTL in an environmental setting have been shown by Qiu et al. (2017) where they utilized a convolutional neural network architecture with hard parameter sharing to forecast rainfall across multiple regions. The MTL objective allowed them to exploit multi-site features. Cirstea et al. (2018) propose a convolutional recurrent neural network with an auto-encoder to forecast correlated time-series in a sensor context. Their MTL set-up consists of a forecasting objective and a reconstruction objective. The latter objective instills the learning of robust features – useful when dealing with noisy sensor data. Mahmoud et al. (2020) propose a systematic approach for MTL in time series and apply it on personalized human activity recognition.

Notably, all MTL approaches developed so far assume (some explicitly, e.g. Mahmoud et al. (2020)) alignment in time of the multiple tasks. However, this assumption breaks down in many real-life scenarios. For example, Jiang et al. (2017) show the difficulties of dealing with mixed-frequency data in GDP forecasting and propose the use of MIDAS regression (Ghysels et al., 2004) preceded by a dynamic factor model. Note, that MIDAS regression assumes a consistent frequency ratio between tasks, making this infeasible for scenarios where a lower frequency tasks is also of an irregular nature. In the domain of water status in fruit crops, ψ_{stem} is typically measured at irregular intervals

of 7 to 14 days, while ψ_{soil} is typically available on a daily or hourly timescale with fixed intervals (Janssens et al., 2011).

By incorporating mixed-frequency data into an MTL framework, it becomes possible to capitalize on the benefits of MTL, even in scenarios where the tasks are only available at different and possibly irregular frequencies. Ultimately, this results in a more flexible and adaptive training process. Recently, the work of Toda et al. (2022) has combined a mixed-frequency objective with a similar – yet different – learning paradigm, namely aggregate learning. They propose a mixed-frequency aggregate learning (MF-AGL) model capable of predicting regional heterogeneity of economic indicators in real-time. However, aggregate learning focuses on updating the model at an aggregate level using a specific set of sources while MTL focuses on learning shared representations across tasks to improve performance on each individual task. Here, we are interested in the latter. To the best of our knowledge, we are the first to combine a mixed-frequency objective with an MTL paradigm.

Finally, note that so far, ψ_{stem} has only been predicted using traditional single-task approaches, with limited success in accurately capturing the dynamic changes over time. As an illustration, Ohana-Levi et al. (2022) use gradient boosted regression trees and González-Teruel et al. (2022) suggest the use of random forests. MTL has never been applied in smart irrigation.

3 Problem Formulation

We consider a multi-task mixed-frequency time-series problem with inputs containing multivariate time-series along with potential metadata, where the goal is to learn a function that performs well on all tasks. For clarity, we limit ourselves to two tasks in what follows. Let $\mathcal{D} = (\mathbf{s}_i, \mathbf{x}_i, \mathbf{y}_i^{(reg)}, \mathbf{y}_i^{(mf)})_{i=1}^n$ be a dataset of n observations, where $\mathbf{x}_i \in \mathbb{R}^{T \times D}$ is a multivariate time-series input with T time-steps and D features, $\mathbf{s}_i \in \mathbb{R}^P$ represents static metadata with P features, $\mathbf{y}_i^{(reg)} \in \mathbb{R}^{H_1}$ is the output for the first task with forecast-horizon H_1, measured at a regular frequency, and $\mathbf{y}_i^{(mf)} \in \mathbb{R}^{H_2}$ is the output for the second task with forecast-horizon H_2 measured at a lower and potentially irregular frequency. Note that, as a result of the mixed frequency between tasks: $\exists\, i : \mathbf{y}_i^{(mf)}$ is undefined.

Our goal is to learn a function $f(\mathbf{S}, \mathbf{X})$ that maps the inputs $\mathbf{X}$ and $\mathbf{S}$ to the outputs $\mathbf{Y}^{(reg)}$ and $\mathbf{Y}^{(mf)}$ such that it performs well on both tasks. More formally, we seek to find the parameters θ of the function $f(\cdot; \theta)$ that minimize the following objective function – Eq. (1):

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \Big(\alpha \cdot \mathcal{L}_{reg}(f(\mathbf{s}_i, \mathbf{x}_i; \theta), \mathbf{y}_i^{(reg)}) + (1 - \alpha) \cdot [y_i^{(mf)} \neq \text{NA}] \mathcal{L}_{mf}(f(\mathbf{s}_i, \mathbf{x}_i; \theta), \mathbf{y}_i^{(mf)}) \Big) \tag{1}$$

where $\mathcal{L}_{reg}$ and $\mathcal{L}_{mf}$ are task-specific loss functions. $[y \neq \text{NA}]$ is an indicator function that returns 1 if y is not missing, and 0 otherwise which allows the training process to deal with data measured at mixed frequencies. The weighting factor α controls the relative importance of the tasks in the optimization problem.

The choice of α is important as it balances the relative contribution of each task in the learning process. In the case of the mixed-frequency task, a larger value[1] of α might be appropriate to ensure that the model is not overly influenced by the less frequently updated data. On the other hand, if the lower-frequency task is more important or contains more informative data, a lower value of α might be more appropriate with $\alpha \in [0, 1]$. Ultimately, the choice of weighting factor can be driven by domain knowledge or can be obtained through a hyperparameter search. Note that Eq. (1) describes the case for two tasks. This can easily be extended to M tasks by adding a term for each task, weighted by a weighting factor β_m, and normalized across tasks by $\sum_{m=1}^{M} \beta_m$.

4 The MultiMix TFT

The MultiMix TFT consists of two main components: (i) the multi-task mixed-frequency framework and (ii) the Temporal Fusion Transformer.

[1] Note that a larger value of α results in a lower weight on the mixed-frequency task.

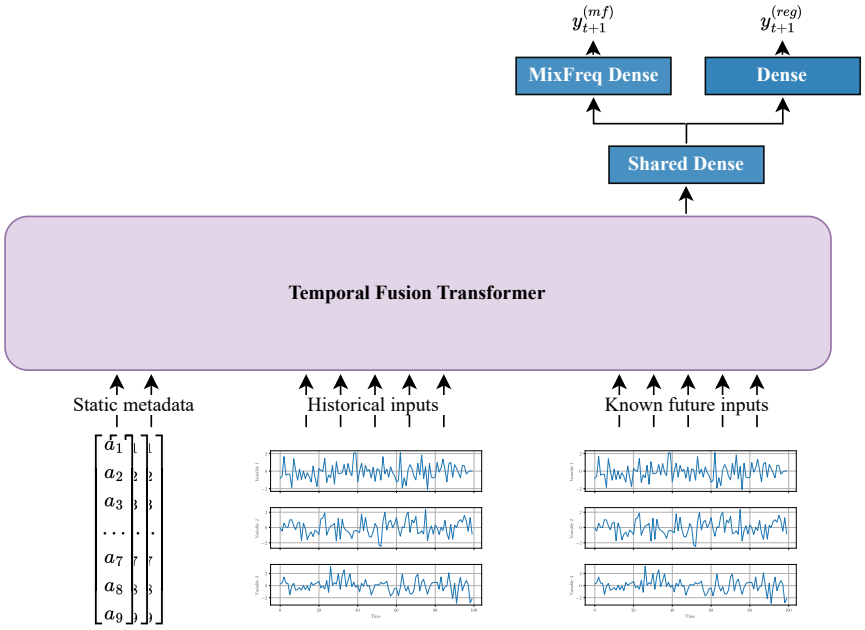


Figure 2: The architecture for the MultiMix TFT. The TFT can take as input: static metadata, historical inputs, and known future inputs (for more details see Lim et al. (2021)). The output of the TFT is then fed to the multi-task mixed-frequency head.

4.1 Multi-task Mixed-frequency Framework

We propose a Multi-task Mixed-frequency (MultiMix) learning approach to modeling time-series data that leverages the relationships between mixed-frequency tasks to enhance the accuracy and robustness of time-series forecasting across tasks. For simplicity, we will explain the framework for two tasks, but this can easily be generalized to m tasks. The approach is particularly useful in situations where one of the tasks is measured at fixed and regular intervals (e.g., every hour) while the other task contains fewer measurements captured at irregular and less frequent intervals. As such, the main goal is to improve the accuracy of the lower-frequency task by leveraging potential cross-task relationships with the higher and regular frequency task.

The key idea behind the MultiMix learning framework is to design a neural network architecture that can handle the different frequencies of the tasks and the interaction between them. To this end, we introduce a MultiMix head which can be placed on top of existing neural network architectures as a final output layer. Note that we use a hard parameter-sharing multi-task model due to its proven efficacy in the context of limited data (Ruder, 2017). Here, we suggest the use of the Temporal Fusion Transformer (Lim et al., 2021) as the base (see Section 4.2) – followed by the MultiMix head. A full schematic overview of the architecture is shown in Figure 2. The outputs of the MultiMix head are then fused into a single objective function (see Eq. (1)).

The presence of mixed-frequency tasks in time-series forecasting poses a unique challenge in multi-task learning, as not all tasks may have corresponding values for each sample within a batch. Traditional multi-task learning training schemes are not equipped to handle such scenarios. Consequently, MultiMix employs a custom training scheme that incorporates a masking mechanism to exclude tasks from the training step when the target value is non-existent when no values are available due to the mixed-frequency nature. Such training scheme is provided in Algorithm 1. Note that the algorithm takes as input – amongst others – a mask for the mixed-frequency task to account for non-existent values due to the lower frequency of said task. Within a given batch, the algorithm first applies the mask to the true values of the low-frequency task and checks if all values within the batch are missing. If so, the loss for this task at the current training step is set to zero. Otherwise, the mask is also applied to the predicted values for the mixed-frequency task for the given batch. As a result, the loss function at the current training step is computed only on the remaining non-missing values for that task. Additionally, a mask for backpropagation at the current training step is computed as the complement of the mask for the mixed-frequency task such that only valid gradients are propagated back through the model during training.

Algorithm 1 Calculate multi-task mixed-frequency loss with masking over a given batch

Input: Predicted values $\hat{y}^{(reg)}$ and $\hat{y}^{(mf)}$, true values $y^{(reg)}$ and $y^{(mf)}$, mask for mixed-frequency task $mask_{mf}$, loss functions $\mathcal{L}_{reg}$ and $\mathcal{L}_{mf}$, weighting factor α
Output: Loss value $\mathcal{L}_{MultiMix}$, $mask_{backprop}$

$y^{(mf)}_{masked} \leftarrow y^{(mf)} \odot mask_{mf}$ ▷ Apply mask to target
if $mask_{mf} = \mathbf{0}$ **then** ▷ All values are missing
    $\ell_{mf} \leftarrow 0$
else
    $\hat{y}^{(mf)}_{masked} \leftarrow \hat{y}^{(mf)} \odot mask_{mf}$ ▷ Apply mask to prediction
    $\ell_{mf} \leftarrow \mathcal{L}_{mf}(\hat{y}^{(mf)}, y^{(mf)})$ ▷ Compute loss on non-missing values
end if
$\ell_{reg} \leftarrow \mathcal{L}_{reg}(\hat{y}^{(reg)}, y^{(reg)})$ ▷ Compute loss on regular task
$\mathcal{L}_{MultiMix} \leftarrow \alpha\mathcal{L}_{reg} + (1-\alpha)\mathcal{L}_{mf}$ ▷ Compute weighted sum of losses (cf. Eq. (1))
$m_{backprop} \leftarrow 1 - mask_{mf}$ ▷ Mask for backpropagation
return $\mathcal{L}_{MultiMix}$, $mask_{backprop}$

4.2 Temporal Fusion Transformer

The Temporal Fusion Transformer is a state-of-the-art deep learning architecture designed specifically for time-series modelling (Lim et al., 2021). It combines the strengths of two well-known neural network architectures: the Transformer (Vaswani et al., 2017) and the Long Short-Term Memory network (LSTM) (Hochreiter & Schmidhuber, 1997). The Transformer architecture is widely acknowledged for its ability to effectively handle sequential data through parallel processing, with applications ranging from text (Vaswani et al., 2017), images (Dosovitskiy et al., 2021), to time-series (Lim et al., 2021). Despite its proficiency, the Transformer architecture has limitations in capturing local patterns that are crucial for precise forecasting of time-series data. Conversely, LSTMs excel at capturing local patterns in sequential data but lack the ability for parallel processing of extended sequences.

The TFT addresses these limitations by incorporating the Transformer and LSTM architectures into a single network. The TFT can include temporal information of historic and known future inputs, as well as static information. The temporal information propagates through the LSTM layer(s) for local extraction after which the resulting embeddings are fed to the transformer component – enriched with static information. This symbiosis allows the TFT to seamlessly extract both local and global patterns from time-series data, while conditioning on static information. Lastly, the TFT contains variable selection networks and gated residual networks (GRNs). The former utilize learned gating mechanisms with softmax to selectively weigh and combine the input features at each time-step, effectively modeling the relevance of each feature and allowing for a measure of variable importance (see also Section 7). Meanwhile, the GRNs equip the TFT with the capacity to adapt its complexity dynamically to a given dataset. These sub-networks play a pivotal role in positioning the TFT as the backbone for the MultiMix framework, particularly due to their inherent capability to accommodate diverse dataset scenarios, ranging from small to large scales, with remarkable flexibility. The results are a noticeable enhancement in forecasting accuracy in different domains ranging from traffic prediction (Zhang et al., 2022) to wind speed prediction (Wu et al., 2022) and ψ_{soil}-prediction (Deforce et al., 2022). For full details on the TFT we refer to the original work of Lim et al. (2021).

5 MultiMix TFT Application to Smart Irrigation

We utilize a challenging dataset obtained from Janssens et al. (2011) containing the two targets of interest: (i) ψ_{soil} and (ii) ψ_{stem} along with many other environmental and plant-physiological variables. The data was collected from 2007-2009. ψ_{soil} is available at daily frequencies, while ψ_{stem} is expensive and complex to measure, resulting in limited data availability captured at irregular intervals and at a lower frequency than ψ_{soil}. To this end, our MultiMix TFT can be used to model both tasks simultaneously, leveraging the higher-frequency ψ_{soil}-information to improve the accuracy of the irregular lower-frequency ψ_{stem}-predictions.

5.1 Soil Water Potential

The ψ_{soil}-measurements were measured by Watermark soil moisture sensors (Irrometer Company, Inc., USA) located in three pear orchards in Belgium. Its values, expressed in pressure units (in kPa), are negative and reflect the strength of the forces holding water in the soil pores. The drier the soil, the more negative the value, and the harder it becomes

Table 1: Overview of variables in the model with their type and a brief description.

Variable	Type	Description
Soil water potential (ψ_{soil})	Target	The ψ_{soil}-values averaged per day and per plot at a depth of 30cm
Stem water potential (ψ_{stem})	Target	The mid-day ψ_{stem}-values per plot measured at irregular time-intervals
Orchard name	Static categorical	Orchard name differentiates between orchards and (implicitly) their characteristics
Soil texture	Static categorical	The texture of the soil at a 0-30cm depth
Pruning treatment	Static categorical	Whether roots were pruned or not
Irrigation treatment	Static categorical	Whether deficit irrigation was applied or not
Measurement year	Static categorical	Measurement year
Measurement month	Temporal known categorical	Measurement month
Precipitation	Temporal known numeric	Daily total precipitation
Reference evapotranspiration	Temporal known numeric	The reference evapotranspiration (ETo), same for all fields here
Relative time index	Temporal known numeric	Time index (in days) since k (see Section 6.1)
Soil water potential	Temporal historic numeric	History of ψ_{soil}
Irrigation amount	Temporal historic numeric	Amount of irrigation applied to a specific plot
Precipitation	Temporal historic numeric	Daily precipitation
Soil temperature	Temporal historic numeric	Daily mean soil temperature around soil moisture sensors (measured by soil moisture sensor)
Reference evapotranspiration	Temporal historic numeric	The reference evapotranspiration (ETo), same for all fields here

for plant roots to extract water. ψ_{soil} was measured every four hours during the months of June to August over three years. The discontinuity in time led to a separate time-series for each year instead of one time-series over the three years. To reduce volatility, the daily average of ψ_{soil} was calculated for all sensors for a given year, yielding time-series with an average length of 80 ψ_{soil}-measurements. Additionally, each field was divided into several plots, each containing six sensors at different depths (30cm, 60cm, and 90cm) to capture information at different levels. Knowing that the sensors at a depth of 30 are most representative for the second task, ψ_{stem}-prediction, the sensors at depth 60 and 90 were discarded and the average per plot across the sensors at depth 30 was calculated.

5.2 Stem Water Potential

The ψ_{stem} was measured on a plot level using a Scholander pressure chamber (Cochard et al., 2001). These measurements were conducted approximately every 10 to 14 days at mid-day during sunny days, providing valuable information about the water status of the plants. The ψ_{stem}, which reflects the water status in the plant stem, is a critical factor in determining plant water stress and, thus, plant growth and productivity (Martínez-Gimeno et al., 2018). The lower-frequency of the ψ_{stem} data – with measurements taken only every 10 to 14 days – compared to the higher-frequency regular ψ_{soil} data, signifies that this is a mixed-frequency problem. This poses a unique challenge in terms of modeling and prediction. However, incorporating the forecast of ψ_{stem} into irrigation management systems can provide a more complete model of the plant water status, contributing to the advancement of precision agriculture.

5.3 Input Variables

The prediction of ψ_{stem} and ψ_{soil} in multiple orchards is influenced by a wide range of variables. Here, we have grouped the variables by plot, and per year – corresponding to the structure of ψ_{stem} and ψ_{soil}. An overview of all variables and their type is provided in Table 1.

6 Training Procedure

The training procedure is a critical component in the development of an accurate forecasting model. In this section, we discuss the measures taken in the training of our forecasting model utilizing the MultiMix TFT architecture (see Figure 2), as well as the benchmarks adopted for comparative analysis.

6.1 MultiMix TFT

At a given time-step t, the objective is to forecast ψ_{soil} and ψ_{stem} for τ_{max} steps ahead using a window of historical observations of size k. This historical information can encompass multivariate exogenous time-series $\mathbf{x}_{i,t-k:t}$ as well as the target variable $\mathbf{y}_{i,t-k:t}$. Note that the history of the lower-frequency task is not included , primarily due to its restricted availability within the selected window of size k. Furthermore, the TFT also allows for the inclusion of known future inputs (numerical or categorical) as $\mathbf{x}_{i,t+1:t+\tau max}$. The TFT architecture can also leverage static information S to enhance its predictive capabilities based on static information such as e.g. orchard location. Hence,

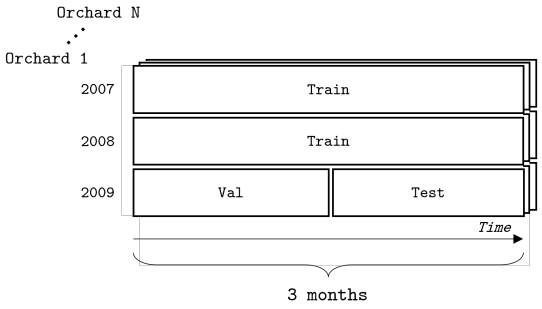


Figure 3: Train-Val-Test split. Note that each year for a given sensor per orchard is considered as a unique time-series as described in Section 5.1.

Table 2: Train-val-test split for each task.

Subset	ψ_{soil}	ψ_{stem}
Train	5576 ($\approx 65\%$)	140 ($\approx 60\%$)
Validation	1254 ($\approx 15\%$)	48 ($\approx 20\%$)
Test	1942 ($\approx 20\%$)	43 ($\approx 20\%$)
Total	8772	231

our goal is to learn a mapping $f(S, \mathbf{x}_{i,t-k:t}, \mathbf{y}_{i,t-k:t}, \mathbf{x}_{i,t+1:t+\tau max}; \theta) = (y^{(reg)}_{t+1:t+\tau_{max}}, y^{(mf)}_{t+1:t+\tau_{max}})$. A detailed account of the input variables and their respective types for each model is provided in Table 1. For our case, $\tau_{max} = 1$ or a one-day ahead forecast, with $k = 7$ or a one-week look-back window as advised by agricultural experts.

Given that ψ_{soil} and ψ_{stem} contain large variations in their measurements, a loss is chosen that can effectively balance the impact of outliers and inliers in the data during training. As such, the Huber loss (Huber, 1964) was chosen for both $\mathcal{L}_{reg}$ and $\mathcal{L}_{mf}$ (cf. Eq. (1)) and is shown in Eq. (2) – Appendix A. Further, we use multi-headed attention[2] as it enables the allocation of at least one attention-head to each task. This allows the model to learn task-specific representations by focusing on the pertinent input information for each task separately.

To ensure a reliable evaluation of the model performance, the dataset is partitioned into a training, validation and test set across tasks. Herein, the temporal and grouped structure of the data was respected in order to prevent information leakage between the training, validation and test sets. A visual representation of the data partitioning is presented in Figure 3. Note that all time-series observations from 2007 and 2008 are exclusively used for training. This guarantees that the model is exposed to two complete growing seasons across years during training, ensuring that the model is able to capture the seasonal variations in the data. Due to the mixed-frequency data, the final sizes of datasets differ for each task as presented in Table 2. Lastly, incorporating temporal and group attributes in data splitting results in non-uniform partitioning of the datasets, thereby producing splits of varying sizes.

To optimize the model parameters, we use the Ranger optimizer (Wright, 2019), a combination of the successful RAdam (Liu et al., 2020) and LookAhead (Zhang et al., 2019). During training, we randomly sample subsequences of length k from each multivariate training sequence along with corresponding static data s and use them as inputs to the model. Hyperparameters of the MultiMix TFT were selected using Bayesian hyperparameter optimization (Bergstra & Bengio, 2012). Bayesian hyperparameter optimization is a model-based approach to hyperparameter optimization that aims to efficiently explore the hyperparameter space and find the set of hyperparameters that optimizes the target metric with demonstrated benefits over its peers such as random search or grid search (Turner et al., 2020). In our case, the target metric was a uniformly weighted sum of the validation MAE (see Eq. (3) – Appendix B) for each task, with each task normalized to a range of [0,1]. By using a uniformly weighted sum across normalized tasks, we prevent Bayesian optimization from favoring one task over the other. We defined a range and/or distribution of

[2]In a multi-horizon forecast, we advise the use of causal attention to prevent information leakage

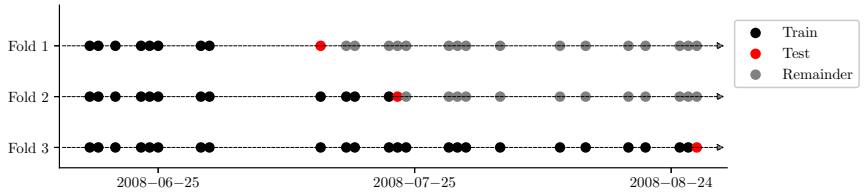


Figure 4: An example of group-aware time-series cross-validation for a single group (i.e. plot) and a single year. The same CV-strategy is then applied across groups to obtain the necessary CV-splits.

hyperparameters to search over, including the learning rate, batch size, number of LSTM layers, dropout rate, and others – for a full overview see Table 4 in Appendix C.2.1. We also included hyperparameters specific to the TFT, such as the size of the continuous hidden layers (e.g. for the LSTM-layers), the size of the hidden layers (e.g. for the variable selection networks – see Section 4.2) and the number of attention heads. While Lim et al. (2021) originally suggested to average across attention heads, we propose a summation across attention heads in an attempt to preserve potential fine-grained information. The underlying rationale for this modification is the conjecture that averaging might inadvertently diminish the granularity of the aggregated representation, whereas summation preserves the individual contributions from each attention head, thereby preserving subtler details which could be useful for the separate tasks. In an endeavor to rigorously examine the efficacy of both techniques, we have incorporated mean and summation as variants of the MultiMiX TFT in our final evaluation, facilitating an exhaustive comparison of their respective influences on the model's performance. A full overview of the hyperparameters, their priors, and the posteriors is shown in Table 4 in Appendix C.2.1. Note that the search space for the network is constrained due to the relatively small size of the data to prevent the generation of excessively large networks during the Bayesian hyperparameter optimization. An early-stopping criterion was used with a patience of 20 epochs after which the training is stopped and the model with the best validation loss is selected. The MultiMix TFT is trained for 150 epochs on a single NVIDIA Tesla V100 GPU. The training takes approximately one hour to complete. The learning curves are presented in Appendix C.1.

6.2 MULTIMIX LSTM

To further emphasize the effectiveness of the TFT as an architectural choice, we allow for a comprehensive evaluation by creating a baseline model using the MultiMix framework in combination with a vanilla LSTM architecture. This baseline model, referred to as the MultiMix LSTM, serves as a point of comparison, illustrating the differences in performance when the MultiMix framework is paired with various architectures. By comparing the MultiMix TFT with the MultiMix LSTM, we aim to highlight the substantial improvement in performance achieved by utilizing the TFT architecture within the MultiMix framework, while demonstrating the limitations of the LSTM in this context. Basic hyperparameters such as the loss function, k, etc. are set to the same values as the MultiMix TFT. Other hyperparameters – as with the MultiMix TFT – are chosen using Bayesian hyperparameter optimization (Bergstra & Bengio, 2012). A full overview of the search space and final hyperparameters is presented in Table 5, Appendix C.2.2. An implementation of the MultiMix TFT and MultiMix LSTM is available at `https://github.com/B-Deforce/multimix_tft`.

6.3 GENERAL BASELINES

Our primary objective is to improve the prediction of ψ_{stem}. However, ψ_{stem} is a challenging variable to predict when considered as a single-task due to limited available data (see also Table 2). As such, the limited data in the single-task context necessitates the use of smaller models as baselines, as opposed to other deep learning models.

We use several well-established methods as baselines, including linear regression and its Lasso variant, decision trees, random forest, gradient boosting, and Gaussian process regression (GPR) (Hastie et al., 2009; Sipper, 2022; Rasmussen & Williams, 2006). Note that GPR was not applied on ψ_{soil} due to its high computational cost and since our interest is primarily in improving ψ_{stem}-prediction. By comparing the performance of our MultiMix TFT model to these baseline models, we can demonstrate its ability to substantially improve ψ_{stem}-predictions while maintaining strong performance on ψ_{soil}. To fine-tune the baseline methods, we employ a group-aware time-series cross-validation (GATS-CV) as shown in Figure 4. GATS-CV preserves the temporal dependencies between data points and across groups, thereby avoiding leakage of future information into the training set. The small size of the baseline models permits to perform a full grid search for hyperparameter tuning. By using GATS-CV with grid search, we ensure that our models are trained and validated on representative subsets of the data, while enabling us to assess the generalization performance of the chosen hyperparameters on unseen data and select the best set of hyperparameters. A full overview of the search space and final hyperparameters is presented in Table 6, Appendix C.2.3. For GPR, a separate overview is presented in Table 7, Appendix C.2.4, with multiple kernels over which the full grid search with GATS-CV is performed.

6.4 PERFORMANCE EVALUATION

To evaluate the final performance of all methods on the unseen test data, the mean absolute error (MAE) and the root mean squared error (RMSE) are computed following Eq. (3) – Appendix B. Furthermore, the mean absolute percentage error (MAPE) and its median counterpart (MdAPE) are also given (see Eq. (4) – Appendix B) to better understand the relative performance. Lastly, Pearson's correlation coefficient is used to assess the linear relationship

Table 3: Comparison of ψ_{stem}- and ψ_{soil}-prediction between MultiMix TFT and single-task baselines.

Model	ψ_{stem}					ψ_{soil}				
	MAE	**RMSE**	**MAPE**	**MdAPE**	**Pearson's Corr**	**MAE**	**RMSE**	**MAPE**	**MdAPE**	**Pearson's Corr**
Linear Regression	0.204	0.064	58.794	42.729	0.499	**0.024**	**0.001**	4.488	**2.463**	**0.986**
Lasso Regression	0.197	0.049	92.993	29.416	0.408	**0.024**	**0.001**	4.592	2.513	**0.986**
Decision Tree	0.193	0.057	100.844	45.453	0.284	0.027	0.002	4.82	2.665	0.984
Random Forest	0.155	0.037	88.994	20.788	0.718	0.035	0.002	7.54	3.478	0.978
Gradient Boosting	0.162	0.041	92.264	24.197	0.444	0.041	0.003	8.732	4.454	0.968
Gaussian Process Reg.	0.197	0.051	98.445	32.372	0.224	-	-	-	-	-
MultiMix LSTM	0.15	0.035	38.158	23.677	0.481	0.047	0.005	8.831	5.324	0.956
MultiMix TFT - Mean	0.093	0.013	20.909	16.767	0.774	0.035	0.003	7.259	3.723	0.975
MultiMix TFT - Sum	**0.078**	**0.009**	**17.226**	**14.736**	**0.859**	**0.024**	**0.001**	**4.244**	2.466	**0.986**

between predictions and truth. These metrics, respectively, give insight into overall absolute performance, larger errors, and relative performance – with MdAPE the robust relative performance – of the model. The final reported metrics are aggregated across all forecasts to obtain global metrics.

7 Results

The experimental results, as summarized in Table 3, reveal that the MultiMix TFT outperforms the baseline models across all performance metrics in the mixed-frequency task of ψ_{stem}-prediction. The MultiMix TFT notably achieves a $\sim$50% improvement in terms of MAE and a $\sim$20% gain in Pearson's correlation compared to the best baseline. Also observe how the gap between MAPE and MdAPE is small for the MultiMix TFT compared to all baseline models. This shows that the MultiMix TFT is consistent and does not make excessively large errors as opposed to the baselines. The gap between sum-based aggregation and mean-based aggregation across attention heads (cf. Section 6.1), indeed suggests that summation is beneficial within the context of the MultiMix framework. It is important to note that the MultiMix LSTM exhibits inferior performance when compared to the MultiMix TFT, highlighting the effectiveness of the latter architecture. While the random forest obtains slightly better results for Pearson's correlation, such performance is not consistently observed across other metrics. Further, GPR demonstrated suboptimal performance, which could be attributed to several factors. For example, the inherent noise present in (agricultural) sensor data may impede the model's ability to discern the true underlying relationships. Moreover, GPR's reliance on a fixed kernel function might limit its capability to accurately capture complex interactions or non-stationary patterns in the multivariate time-series data. We note that GPR could potentially improve with further kernel engineering (beyond the current kernels considered in Appendix C.2.4). Upon further investigation, almost all baselines show strong signs of overfitting, highlighting the difficulty of ψ_{stem}-prediction in relation to the available data. We'd like to emphasize that the MultiMix TFT does not suffer from this issue thanks to the hard parameter sharing with ψ_{soil}-prediction. As such, these findings provide compelling evidence that the MultiMix TFT successfully leverages the concurring regular-frequency task, ψ_{soil}, to improve the prediction of the mixed-frequency task of ψ_{stem} prediction, thereby establishing its superiority over the baseline models.

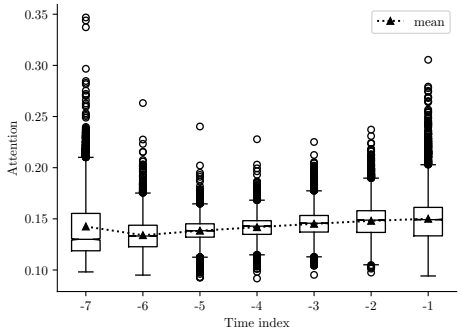


Figure 5: Boxplot of means across all three attention heads of the MultiMix TFT for all training samples

After examining the results of ψ_{soil}, it initially appears that both the baseline models and the MultiMix TFT obtained strong skills. However, upon further analysis, it becomes apparent that there exists a high correlation between the last known value of ψ_{soil} (i.e. $\psi_{soil,t-1}$) and $\psi_{soil,t}$, thereby suggesting that all models essentially learn a trivial naive forecast as the optimal model for ψ_{soil}-prediction. Nonetheless, while the MultiMix TFT exploits this information as well, it simultaneously succeeds in learning a representation that is both useful for predicting ψ_{soil} as well as ψ_{stem}. This is also illustrated in Figure 5 where we visualized the average attention across all attention heads. From this, we derive that the MultiMix TFT pays attention to more than just the last known value. Furthermore, it is important to note that baselines which learn a trivial naive forecast may not be generalizable when expanding the forecasts to a multi-horizon setting, due to its inherent simplicity and lack of consideration for the complexities of the underlying data.

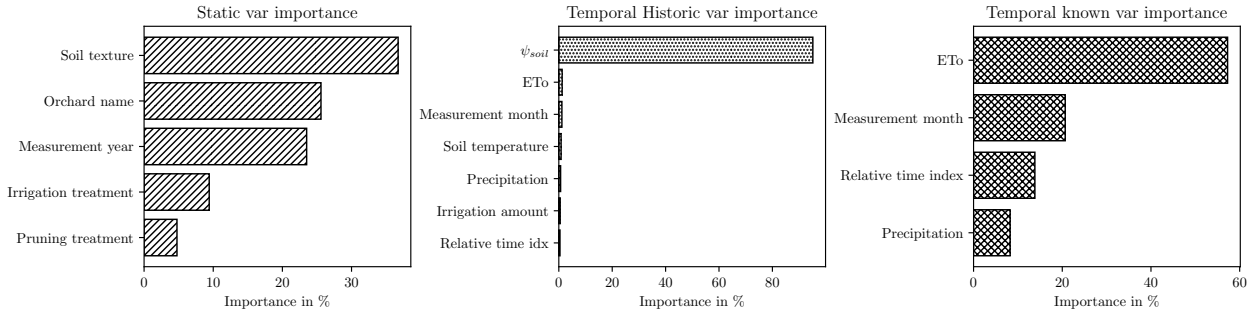


Figure 6: Overview of the variable importance obtained through the variable selection networks from the TFT

In contrast, the underlying TFT is known to perform well in a multi-horizon setup, owing to its ability to effectively model and account for the intricacies of the multivariate data (Lim et al., 2021).

Figure 6 shows the variable importance obtained from the variable selection networks (see Section 4.2). Note that each *type* of data (i.e. static metadata, temporal historic inputs, temporal known inputs) shares a unique set of weights per variable across time-steps. As a result, we obtain three distinct (one per data *type*) variable importance overviews scaled to [0,1]. We note resemblance with expert knowledge as soil texture and orchard name both receive high importance as static metadata. Intuitively, this makes sense as orchard name and soil texture are essentially a proxy for orchard-specific characteristics. For historical information, the disproportionate importance of ψ_{soil} stands out. Presumably, this is a result of the model exploiting the high correlation between $\psi_{soil,t}$ and $\psi_{soil,t-1}$ for predicting ψ_{soil}. ψ_{soil} – a measure of soil moisture (cf. Section 5.1) – is of course also a good indicator for plant water stress ψ_{stem}. Interestingly, the runner-up – ETo – corresponds to domain knowledge as well. ETo can be considered a proxy for water demand. Hence, the combination of ψ_{soil} and ETo can act like a supply-and-demand mechanism. Lastly, the known future inputs also correspond well with expert knowledge. As such, these observations confirm that the TFT – underlying the MultiMix framework – effectively aligns with expert knowledge.

8 Conclusion

We proposed the MultiMix framework, a novel approach for multi-task mixed-frequency learning for time-series forecasting. This framework addresses the challenge of forecasting multiple tasks containing different and/or irregular measurement frequencies. To do this, the MultiMix framework uses hard parameter sharing to capture the commonality between the tasks, followed by a customized MultiMix head that can effectively deal with mixed-frequency data during training through a masking mechanism and custom loss function.

We implemented the framework using the state-of-the-art TFT, and applied it to a real-world dataset in the agricultural domain. Specifically, the framework was used to forecast ψ_{soil} and ψ_{stem}, two tasks that are measured at different frequencies. Our experimental results showed that our proposed approach outperforms several state-of-the-art baselines in terms of various metrics, especially for the task with the lower measurement frequency, i.e. ψ_{stem}. This demonstrates the effectiveness of our proposed framework in a setting with multiple tasks at hand, where one of the tasks is measured at a lower and/or irregular frequency. The variable selection networks, inherent to the TFT, show resemblance with expert knowledge. However, due to their location in the TFT, they cannot be interpreted for each task separately. An interesting future direction would be to adapt the TFT so that each variable selection network can be assigned to one task.

We believe that our proposed framework can be applied to a wide range of domains, where data is collected at different frequencies, such as e.g. finance, energy, and health. As such, exploring the generalizability of the approach on additional datasets from diverse domains can provide further insights into its applicability. In this work, the framework was only applied on a dual task problem. To further test the scalability of the model, testing it in a set-up with more than two tasks would be beneficial. Lastly, extending the framework for multi-horizon forecasts can be a useful addition for some domains such as e.g. economic forecasting, smart agriculture, and more.

Acknowledgements

Funding: This work was supported by the Research Foundation - Flanders (FWO) [grant number G0C6721N].

REFERENCES

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5799–5811, November 2021.

Mohammed Al-Rawi and Ernest Valveny. Compact and efficient multitask learning in vision, language and speech. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 2933–2942, 2019. doi: 10.1109/ICCVW.2019.00355.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13: 281–305, 2012. doi: 10.5555/2503308.2188395.

Z. Chen, J. E, X. Zhang, H. Sheng, and X. Cheng. Multi-task time series forecasting with shared attention. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pp. 917–925. IEEE Computer Society, nov 2020.

Razvan-Gabriel Cirstea, Darius-Valer Micu, Gabriel-Marcel Muresan, Chenjuan Guo, and Bin Yang. Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 1527–1530, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3269310.

Hervé Cochard, Sébastien Forestier, and Thierry Améglio. A new validation of the Scholander pressure chamber technique based on stem diameter variations. *Journal of Experimental Botany*, 52(359):1361–1365, 06 2001. ISSN 0022-0957.

Boje Deforce, Bart Baesens, Jan Diels, and Estefanía Serral Asensio. Forecasting sensor-data in smart agriculture with temporal fusion transformers. In *Transactions on Computational Science & Computational Intelligence*. Las Vegas (USA), Springer Nature, 2022.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–16, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 4174–4182. IEEE Computer Society, 2017.

Eric Ghysels, Pedro Santa-Clara, and Rossen Valkanov. The midas touch: Mixed data sampling regression models. 2004.

Juan D. González-Teruel, Maria Carmen Ruiz-Abellon, Víctor Blanco, Pedro José Blaya-Ros, Rafael Domingo, and Roque Torres-Sánchez. Prediction of water stress episodes in fruit trees based on soil and weather time series data. *Agronomy*, 12(6), 2022. ISSN 2073-4395. doi: 10.3390/agronomy12061422.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848570.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi: 10.1214/aoms/1177703732.

Pieter Janssens, Tom Deckers, Frank Elsen, Annemie Elsen, Hilde Schoofs, Wim Verjans, and Hilde Vandendriessche. Sensitivity of root pruned 'conference' pear to water deficit in a temperate climate. *Agricultural Water Management*, 99(1):58–66, 2011. ISSN 0378-3774.

Yu Jiang, Yongji Guo, and Yihao Zhang. Forecasting china's gdp growth using dynamic factors and mixed-frequency data. *Economic Modelling*, 66:132–138, 2017. ISSN 0264-9993. doi: https://doi.org/10.1016/j.econmod.2017.06.005.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.03.012.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Reem A. Mahmoud, Hazem Hajj, and Fadi N. Karameh. A systematic approach to multi-task learning from time-series data. *Applied Soft Computing*, 96:106586, 2020. ISSN 1568-4946.

M.A. Martínez-Gimeno, L. Bonet, G. Provenzano, E. Badal, D.S. Intrigliolo, and C. Ballester. Assessment of yield and water productivity of clementine trees under surface and subsurface drip irrigation. *Agricultural Water Management*, 206:209–216, 2018.

Noa Ohana-Levi, Igor Zachs, Nave Hagag, Liyam Shemesh, and Yishai Netzer. Grapevine stem water potential estimation based on sensor fusion. *Computers and Electronics in Agriculture*, 198:107016, 2022. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2022.107016.

Minghui Qiu, Peilin Zhao, Ke Zhang, Jun Huang, Xing Shi, Xiaoguang Wang, and Wei Chu. A short-term rainfall prediction model using multi-task convolutional neural networks. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 395–404, 2017.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint*, abs/1706.05098, 2017.

Moshe Sipper. High per parameter: A large-scale study of hyperparameter tuning for machine learning algorithms. *Algorithms*, 15(9), 2022. ISSN 1999-4893.

Takamichi Toda, Daisuke Moriwaki, and Kazuhiro Ota. Aggregate learning for mixed frequency data. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 4157–4165, 2022. doi: 10.1109/BigData55660.2022.10020900.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In Hugo Jair Escalante and Katja Hofmann (eds.), *NeurIPS 2020 Competition and Demonstration Track, 6-12 December 2020, Virtual Event / Vancouver, BC, Canada*, volume 133 of *Proceedings of Machine Learning Research*, pp. 3–26. PMLR, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Less Wright. Ranger - a synergistic optimizer. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer, 2019.

Binrong Wu, Lin Wang, and Yu-Rong Zeng. Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy*, 252:123990, 2022. ISSN 0360-5442.

Wendong Yang, Zhirui Tian, and Yan Hao. A novel ensemble model based on artificial intelligence and mixed-frequency techniques for wind speed forecasting. *Energy Conversion and Management*, 252:115086, 2022. ISSN 0196-8904.

Hao Zhang, Yajie Zou, Xiaoxue Yang, and Hang Yang. A temporal fusion transformer for short-term freeway traffic speed multistep prediction. *Neurocomputing*, 500:329–340, 2022. ISSN 0925-2312.

Michael R. Zhang, James Lucas, Jimmy Ba, and Geoffrey E. Hinton. Lookahead optimizer: k steps forward, 1 step back. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9593–9604, 2019.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022. doi: 10.1109/TKDE.2021.3070203.

A HUBER LOSS

$$\mathcal{L}_{Huber}(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2, & \text{for } |\hat{y} - y| \leq \delta \\ \delta(|\hat{y} - y| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \tag{2}$$

Where δ represents the threshold parameter which is obtained through hyperparameter tuning.

B EVALUATION METRICS

$$\text{MAE} = \frac{\sum_{t=1}^{\tau_{\max}} |y_{i,t} - \hat{y}_{i,t}|}{\tau_{\max}} \qquad \text{RMSE} = \sqrt{\frac{\sum_{t=1}^{\tau_{\max}} (y_{i,t} - \hat{y}_{i,t})^2}{\tau_{\max}}} \tag{3}$$

$$\text{MAPE} = \frac{100\%}{\tau_{\max}} \sum_{t=1}^{\tau_{\max}} \left| \frac{y_{i,t} - \hat{y}_{i,t}}{y_{i,t}} \right| \qquad \text{MdAPE} = 100\% \cdot \text{median} \left(\left| \frac{y_{i,t} - \hat{y}_{i,t}}{y_{i,t}} \right|, ..., \left| \frac{y_{i,\tau_{\max}} - \hat{y}_{i,\tau_{\max}}}{y_{i,\tau_{\max}}} \right| \right) \tag{4}$$

Where $y_{i,t}$ represents the measurement for a given task at time t and τ_{max} the length of the forecasting horizon. $\hat{y}_{i,t}$ is the estimated forecast by the model.

C DETAILS OF TRAINING

Below, we provide some extra details on the training process for different models.

C.1 MULTIMIX TFT - LEARNING CURVE

Figure 7 shows the learning curves of the final selected MultiMix TFT for the training and validation set. The best model was chosen based on the validation loss, which was after 30 epochs.

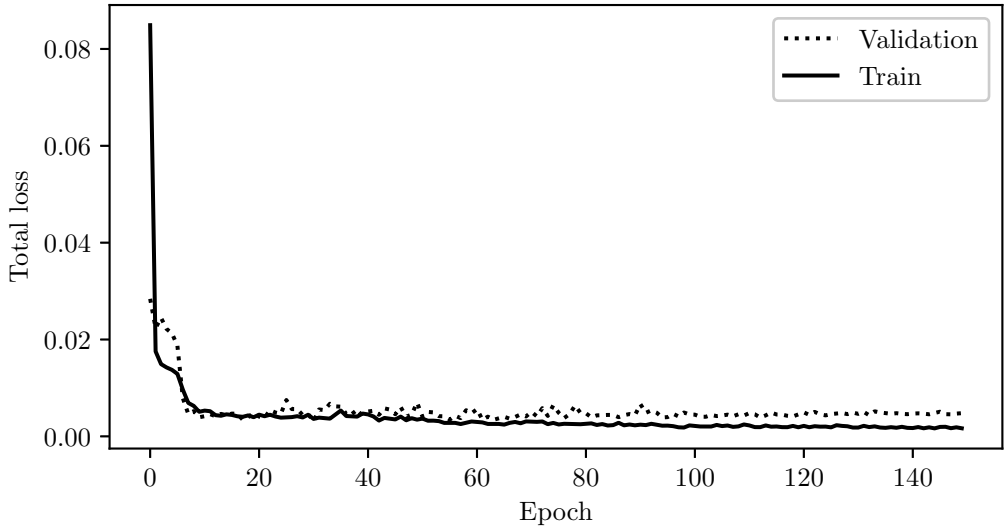


Figure 7: Learning curves of the final selected MultiMix TFT

C.2 HYPERPARAMETER SET-UP

C.2.1 MULTIMIX TFT

Table 4: MultiMix TFT hyperparameters, search spaces for Bayesian hyperopt, and final selection

Hyperparameter	**Prior**	**Final**
Number of Attention Heads	2, 3, 4	3
Aggregation Method	Mean, Sum	Sum
Dropout Rate	Uniform(0.1, 0.8)	0.489
Hidden Continuous Size	8, 16	8
Hidden Size	8, 16	16
LSTM Layers	1, 2, 3	3
Batch Size	32, 64, 128	128
Learning Rate	LogUniform(1e-5, 1e-1)	0.039
$\delta_{\mathcal{L}_{reg}}$ and $\delta_{\mathcal{L}_{mf}}$, see Eq. (2)	Uniform(0.05,0.8)	0.451 – 0.194
α, see Eq. (1)	Uniform(0.05, 0.95)	0.766

C.2.2 MULTIMIX LSTM

Table 5: MultiMix LSTM hyperparameters, search spaces for Bayesian hyperopt, and final selection

Hyperparameter	**Prior**	**Final**
Batch Size	32, 64, 128	32
Dropout Rate	Uniform(0.1, 0.8)	0.696
Hidden Size	8, 16, 32	16
Learning Rate	LogUniform(1e-5, 1e-1)	0.0003
Number of Layers	1, 2, 3, 4	2
$\delta_{\mathcal{L}_{reg}}$ and $\delta_{\mathcal{L}_{mf}}$, see Eq. (2)	Uniform(0.05, 0.8)	0.174 – 0.364
α, see Eq. (1)	Uniform(0.05, 0.95)	0.719

C.2.3 GENERAL BASELINES

Table 6: Other baseline hyperparameters, search spaces for full grid search, and final selection

Model	**Hyperparameters**	**Values**	**Final**
Lasso	α	0.0001-0.1	0.0011
Decision Tree	Depth	3, 5, 7, 9, 11	11
	Min samples split	1, 3, 5, 10, 15, 20	15
	Min samples leaf	1, 2, 4, 8, 16	2
Gradient boosting	N estimators	5, 10, 15, 25, 50, 75	25
	Depth	2, 3, 4, 5, 10	5
	Min samples split	2, 3, 4, 5	5
	Min samples leaf	1, 2, 3, 4, 5	2
	Max features	Sqrt, log2	Sqrt
Random forest	N estimators	5, 10, 15, 25, 50, 75	75
	Depth	2, 3, 4, 5, 10	4
	Min samples split	2, 3, 4, 5	2
	Min samples leaf	1, 2, 3, 4, 5	3
	Max features	Sqrt, log2	log2

C.2.4 GAUSSIAN PROCESS REGRESSION

After a full grid search on the hyperparameters and kernels presented in Table 7, the following kernel was selected:

$$0.316^2 * \text{Matérn}(\texttt{length\_scale} = 5.6, \nu = 0.5)$$

Table 7: GPR hyperparameters, Kernel functions, and search space for full grid search.

Kernel	**Hyperparameters**	**Values**
RBF	Length Scale	0.1, 1.2, …, 10
Matérn	Length Scale	0.1, 1.2, …, 10
	ν	0.5, 1.5, 2.5, 3.5
WhiteKernel	Noise Level	0.1, 0.2, …, 1
ConstantKernel * RBF	Constant Value	0.1, 1.2, …, 10
	Length Scale	0.1, 1.2, …, 10
ConstantKernel * Matérn	Constant Value	0.1, 1.2, …, 10
	Length Scale	0.1, 1.2, …, 10
	ν	0.5, 1.5, 2.5, 3.5
DotProduct	σ_0	0.1, 1.2, …, 10
ExpSineSquared	Length Scale	0.1, 1.2, …, 10
	Periodicity	1, 2, …, 10
RationalQuadratic	Length Scale	0.1, 1.2, …, 10
	α	0.1, 1.2, …, 10
ConstantKernel * RationalQuadratic	Constant Value	0.1, 1.2, …, 10
	Length Scale	0.1, 1.2, …, 10
	α	0.1, 1.2, …, 10