**Creating Effective Time Series for AI Applications: Assessing Minimum Data Requirements.**

Eduart Murcia1 and Sandra M. Guzmán, Ph.D.1

1Indian River Research and Education Center, University of Florida.

1. **Introduction**

Artificial intelligence, specifically Machine Learning (ML) applications, have emerged as essential modeling tools for scientists, technicians, and decision-makers seeking insights into complex processes, particularly those with interrelated variables, such as natural resources management. ML has also become more popular for practical decision-making when other modeling tools are unavailable. Despite the widespread applicability of ML for various applications, determining the minimum amount, granularity, and quality of data is not always straightforward. This publication provides an overview of the factors that should be considered when employing ML applications with Time Series (TS) data as input. Although the information provided in this publication could be relevant for projects using images as input data, these require additional checks that should be performed to address data quality. For more examples of object detection AI applications, refer to the EDIS publication AE529, Applications of Artificial Intelligence for Precision Agriculture (<https://edis.ifas.ufl.edu/publication/AE529>).

1. **How much data should I collect to use ML**?

As ML models are data-driven, having more observations is more beneficial. More observations facilitate accurately capturing patterns, trends, and variations in TS data. While there is no established minimum threshold for the number of observations required to use ML, several factors should be considered including:

**1. Data granularity and complexity:** Collecting data frequently would increase the size of the TS dataset. However, it is essential to consider that higher-frequency data collection might be costly and may not necessarily add value to ML forecasting. As more data is collected, the complexity of ML models required to process it also increases. The minimum number of data points in a TS can be related to the number of parameters an ML must consider. TS with multiple cycles or repetitions would be preferable to those derived from multiple sites but with only one cycle or repetition. For instance, in an irrigation management experiment where data is collected using soil moisture sensors for one month with three irrigations per week, collecting data every minute might not improve ML forecast compared to data collected every 15- 30 minutes. However, TS with a higher number of irrigation events, such as data spanning one year, could potentially improve the forecast. Furthermore, consider the scenario where TS data is obtained from 12 sensors at different locations. Although it provides the same amount of data as a single time series of 12 months, it might present distinct challenges for ML forecasting. Figure 1 shows a TS with data collected at different frequencies or timesteps. For the TS with more frequent data collection (upper graph in Fig 1), identifying trends per watering cycle requires more complex ML models. Conversely, The TS with lower frequency but more irrigation cycles shows more visible trends.

A graph of a graph

Description automatically generated

A graph of a graph

Description automatically generated

Figure 1. Comparison of volumetric water content time series collected from a single soil moisture sensor (1 year). Upper panel: data points collected every hour. Lower panel: data points collected every day.

**2. Machine Learning Algorithm complexity:** Different ML algorithms have different data requirements. Some algorithms require more data to generalize effectively, while others can perform well with smaller datasets. Deep learning algorithms, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), typically require larger datasets due to the number of parameters that need to be learned. However, they are capable of learning more complex structures. On the other hand, simpler algorithms, such as some decision trees with simple parameters, require less data but are usually limited to handling simpler data structures (Brownlee, 2018, 2022). Before selecting the ML, a review of published literature could help guide the most common ML used for the process studied and their respective data requirements. When there is limited data, splitting TS sets for ML training, validation, and testing can be challenging. Insufficient data for validation can result in overfitting, where the model performs well on training data but fails to generalize to new data. Coming back to the example with the soil moisture sensors given in the previous section, training the model with data collected hourly for only one month (i.e., three weeks of data for training and one week for testing) might mislead and let it think that an extreme event that rarely occurs is expected every week. Figure 2 shows an overview of data management for an ML application. Such protocols include data-splitting (where an 80-20% training-testing ratio is commonly used), cross-validation (which for TS is a recursive splitting of the training set into subsets of training and validation using either rolling windows or expanding windows), and benchmarking are necessary for an appropriate ML forecast. Limited data can make challenging these protocols.

A diagram of a training

Description automatically generated

Figure 2. A flow chart of the ML training and testing processes for forecasting tasks. The flow chart shows the overall data split requirements for these processes.

1. **Steps to follow to assess the minimum TS data required.**

Due to the factors mentioned above, standardizing the minimum number of data points is challenging. However, the following steps can be followed to assess if the data collected is sufficient for an ML application:

* Although it may seem intuitive, using your current knowledge is the first step to assessing if the data is sufficient and representative. Making a data graph and conducting an initial TS analysis can save time and potentially enhance the quality of the forecast. A simple graph can reveal valuable data structure information, such as trends, seasonal cycles, and the signal-to-noise ratio (Figure 3). Simple noise-reduction techniques such as a moving average (iterative average calculation) or using low-pass filters (setting cutoff limits to extreme values) are usually helpful; for more details on TS noise reduction, refer to Kostelich and Schreiber (1993) and Premanode et al. (2013). Also, make sure that the length of the TS captures multiple seasonal patterns and at least a change in the TS trend. Figure 3 displays an example of a TS with multiple changes in seasonal patterns and long-term relationships (redlines). In addition, auto-correlation function plots (ACF), used to identify correlations between TS, are easy-to-implement tools that better understand the TS data[[1]](#footnote-3). Figure 3 shows how general checks can be made using graphs. In this example, there are at least 3 short-term cycles in the TS (indicated by red lines) after removing the noise components of the TS (represented by the blue line).

A graph with red lines and numbers

Description automatically generated

Figure 3. Example of a Volumetric Water Content TS being visually analyzed to determine how appropriate it would be to use the data to train an ML forecasting algorithm.

* Consider the length of the ML forecast or how far into the future the prediction is expected to be. Generally, ML forecasts usually cover a range from a few days to a week. Defining the expected time range of the forecast will contribute to understanding how many input data points could be relevant for the ML and how many training sets can be obtained given the length of past and forecast windows.
* Classical statistical forecasting models, such as autoregression or autoregressive integrated moving average, could be an option. It has been found that classical statistical methods can outperform ML algorithms when a few observations are available and the nonlinearities of the TS are not dominant (Cerqueira et al., 2019).
* Use heuristics (rules-of-thumb) as a reference (only as a reference). For statistical algorithms, a preferable minimum number of observations is 100 for training [Box and Tiao, 1975]. Thus, considering ML algorithms require more data than statistical algorithms, expect to factor this number for at least one order of magnitude.

1. **Case study: Assessing Data Needs for Water Forecasting**

In this example, managers are interested in applying ML to forecast Volumetric Ion Content (VIC) and soil water estimates for the next **24 hours** for lettuce, sweet potato, and lemon irrigation. They have been collecting sensor data for four months and want to know whether this data would be enough to train an ML forecasting algorithm that could provide irrigation managers with sufficient time to respond to undesirable conditions.

Data conditions:

Ten soil moisture sensors per crop measured volumetric water content (VWC) and VIC every 15 minutes. The sensors were installed four months ago at the beginning of the season. They have also been continuously collecting physiological data, including chlorophyll content and transpiration rate.

**Lettuce:**

Using expert knowledge, managers know that one entire lettuce season lasts between 30 to 70 days, meaning the TS would have at least two crop seasons. They also know that the crop physiology and soil moisture content is a two-way interaction in which the crop's stage influences the water consumption rate, the soil structure, and the concentration of salts in the root system and the soil. Thus, they know the available data would be enough to train the model with data from multiple short- and long-term variations. Then, they proceeded with graphing the data and performing an initial TS analysis, visual and statistical assessments of the data, and they found the data still seem valid. With this quick analysis, they have decided that this dataset is potentially usable with ML or deep learning algorithms. Finally, they also recognize the importance of benchmarking the results from ML algorithms to guarantee that the proposed approach improves the current state of the art. Thus, they decided to include in the list of algorithms to try a couple of statistical forecasting algorithms.

**Sweet potato:**

Using expert knowledge, managers know that one sweet potato season ranges between 3 to 5 months, and consequently, they start considering that the data might not be enough to be used with complex ML algorithms. However, they acknowledged the high frequency of data collection (15 minutes) that gives them around 11500 data points per sensor. In this scenario, the data size could be worth trying (with the caveat that the complex models could tend to overfit). Thus, after using visual help and performing statistical analysis of the data, they decide not to discard the idea of using ML; however, they have clarity that they would have to limit the model's scope to represent primarily short- and middle-term variations. Also, they decided to use simpler algorithms such as XGBOOST and Light-XGBOOST (Chen and Guestrin, 2016), as based on a literature review, there is evidence of their capacity to perform well in TS forecasting tasks without requiring massive datasets. They realized that testing the model's accuracy with unseen data would not be possible until additional data was collected. For this case, ML cannot be implemented; however, classical statistics might be an option.

**Lemon:**

Using expert knowledge, managers know that a season can last between 3 to 6 years. Therefore, four months of data would not be enough to represent the middle- and long-term variations of the VWC TS. Thus, that data would not be enough for a forecasting system for irrigation management.

**5. Options to eS**

(such as

Concluding remarks

ML algorithms are data-driven approaches whose performance highly depends on the amount of data available. However, data can often be limited in many projects. To obtain meaningful predictions, it is essential to consider multiple factors, including data granularity, complexity, and the complexity of the ML algorithm. This document provides a set of steps for practitioners, researchers, and any scientist interested in starting to use ML, particularly for TS data. This set of steps includes using expert knowledge, visual and statistical tools to characterize the TS, classical statistical approaches to benchmark the ML algorithms and as an alternative when little data is available, and rules-of-thumb as a first filter to determine whether there is enough data. Other approaches to increase the dataset size (data augmentation) were also introduced as alternatives for insufficient data.

**Bibliography**

Brownlee, J. (2018). *How to Develop a Skillful Machine Learning Time Series Forecasting Model*. 1–16. https://machinelearningmastery.com/how-to-develop-a-skilful-time-series-forecasting-model/

Brownlee, J. (2022). How Much Data Is Required for Machine Learning? *Postindustria*. https://postindustria.com/how-much-data-is-required-for-machine-learning/

Cerqueira, V., Torgo, L., & Soares, C. (2019). *Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters*. <http://arxiv.org/abs/1909.13316>

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. <https://doi.org/10.1145/2939672.2939785>

Manu, J. (2022). *Modern Time Series Forecasting With Python: explore industry-ready time series forecasting using modern machine learning and deep learning*.

Molnar, C. (2022). *Modeling Mindsets : the many cultures of learning from data*.

Um, T. T., Pfister, F. M. J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., & Kulic, D. (2017). Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. *ICMI 2017 - Proceedings of the 19th ACM International Conference on Multimodal Interaction*, *2017-Janua*, 216–220. <https://doi.org/10.1145/3136755.3136817>

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2021). Time Series Data Augmentation for Deep Learning: A Survey. *IJCAI International Joint Conference on Artificial Intelligence*, 4653–4660. <https://doi.org/10.24963/ijcai.2021/631>

1. . In the following link there is a Python implementation of an ACF plot with an example that can be used for reference: https://github.com/SIHLAB. [↑](#footnote-ref-3)