ISSN: 2993-8643

Research Article

Engineering: Open Access

A Data Centric Approach Towards Server Time Series Model Compression

Mantej S. Gill[1*], Anil B[2], S. Dhamodhran[3] and Madhusoodhana Chari S[4]

[1]*Hewlett Packard Enterprise Bangalore, India*

[2]*Hewlett Packard Enterprise Bangalore, India*

[3]*Hewlett Packard Enterprise Bangalore, India*

[4]*Hewlett Packard Enterprise Bangalore, India*

[*]Corresponding Author
Mantej S. Gill, Hewlett Packard Enterprise Bangalore, India

Submitted: 2023, Nov 29; **Accepted**: 2024, Jan 05; **Published**: 2024, Jan 10

Citation: Gill, M. S., Anil, B., Dhamodhran, S., Chari S, M. (2024). A Data Centric Approach Towards Server Time Series Model Compression. *Eng OA, 2*(1), 6-12.

Abstract

Effective model compression plays a pivotal role in mitigating the computational and interpretational challenges inherent in the domain of time series forecasting. In this study, we introduce an innovative data-centric methodology tailored to identify a representative data subset from the entirety of the dataset. This chosen representative segment forms the cornerstone for the training of proficient time series forecasting models. Furthermore, our investigation unveils a compelling outcome of this approach—a substantial reduction in the size of time series forecasting models when trained with this selected representative data segment. This model compression strategy results in a remarkable 56.31% decrease in storage consumption, a discovery of considerable significance for optimizing resources and enhancing scalability in time series forecasting. By distilling the dataset to its fundamental components through our data-centric approach, we aim to enhance both computational efficiency and the interpretability of the resultant models. This paper introduces a pioneering technique to tackle the challenges associated with data volume and model complexity in the field of time series forecasting, offering potential pathways for more efficient and insightful modelling in this domain.

Keywords: Time-Series · Model Compression · Data-Centric AI · Representative Segments

Introduction

Time-series forecasting serves as a fundamental component in diverse domains, such as finance, healthcare, environmental monitoring, and industrial processes Rasheed et al. [2010]. This methodology unveils historical data patterns and empowers the anticipation of future trends, rendering it an indispensable tool for informed decision-making. However, as the volume of time-series data continues to grow exponentially in our increasingly interconnected world, the computational and interpretative challenges associated with handling these vast datasets and the corresponding complex models have become increasingly pronounced Ana Almeida and Pinto [2023]. One key challenge in time-series analysis is the efficient utilization of computational resources Shu et al. [2014]. Traditional time series models often require significant storage and computing power, hindering their scalability and real-time applicability Yu and Xiao [2022]. Moreover, as these models become more complex to capture intricate temporal dependencies, they also become less interpretable, potentially limiting their utility in scenarios where interpretability is essential.

Use footnote for providing further information about author (webpage, alternative address)—not for acknowledging funding agencies. In response to these challenges, model compression has emerged as a critical research area aimed at reducing the computational burden of time series analysis without sacrificing predictive accuracy Zhu and Gupta. Existing approaches to model compression typically focus on algorithmic or architectural modifications. While these approaches have shown some success, they often do not fully address the underlying issue of excessive data volume in time-series datasets. To address this, a data-centric approach towards time series model compression has gained attention in recent years. A data-centric approach towards time series model compression Yin et al. [2022] offers a promising solution to these challenges. By prioritizing data reduction and representation techniques, a data-centric approach focuses on compressing the time-series data itself instead of solely relying on modifying the models or algorithms. This approach recognizes that the sheer volume of time-series data is a significant contributor to the computational burden and seeks to efficiently store and process this data while maintaining its essential features and predictive capabilities. One proposed method for effective storage of time-series data is through a novel approach that reduces the computation expense Bhattarai et al. [2019].

This method leverages data compression techniques to reduce the storage requirements of time-series data, allowing for more efficient utilization of computational resources Ryabko [2012]. By compressing the time-series data, this approach enables faster data retrieval and analysis, leading to improved scalability and real-time applicability of time-series models Bhattarai et al. [2019]. Additionally, this data-centric approach also acknowledges the importance of preserving the interpretability of time series models. By reducing the data volume while retaining the essential features of the time series, interpretable insights can still be derived from the compressed data. Furthermore, a data-centric approach recognizes that time series analysis often involves working with specific temporal ranges of data rather than the entire dataset Barez et al. [2023]. To address this, compression and deletion strategies can be implemented to store and retrieve only the relevant portions of the time-series data. One example of a data-centric approach towards time series model compression is the proposed symbolic representation of time series. This symbolic representation allows for a reduction in dimensionality and numerosity, effectively compressing the time-series data.

Continuing with this concept, in this paper, we present a data-centric approach designed to tackle the challenges associated with model compression in the realm of time series analysis. Our methodology is rooted in the idea that not all data points in a time-series dataset are equally informative or necessary for training effective models. Instead, we advocate for the identification of a representative data segment from the complete dataset, which serves as the cornerstone for training time series models. Our research introduces an intriguing outcome of this data-centric approach: a substantial reduction in the size of time series models when trained with the selected representative data segment. By distilling the dataset to its essential components, we aim to significantly reduce storage consumption. In our experimental results, we observe a remarkable 56.31% reduction in storage requirements, a finding of profound significance for resource optimization and scalability in time series analysis. Furthermore, our data-centric approach not only addresses the computational challenges but also enhances the interpretability of the resultant time series models. By focusing on the most informative data, we aim to simplify model complexity, making it easier to understand the underlying patterns and relationships within the data.

In summary, this paper presents a pioneering method to tackle the issues of data volume and model complexity in time series analysis. Our data-centric approach promises more efficient and insightful modelling in this domain, offering potential avenues for improving both computational efficiency and interpretability in time-series analysis. Through a comprehensive exploration of our methodology and experimental results, we aim to demonstrate the transformative potential of this approach in the field of time-series analysis.

Related Works

Previous research has explored various strategies for addressing the challenges of data volume and model complexity in time series analysis. One common approach is pruning, which involves removing unnecessary connections or parameters from a time series model Wielgosz [2020]. This approach aims to reduce the size and complexity of the model by selectively removing components that have little impact on the overall performance. Another strategy is compressing larger models, which involves applying compression algorithms to reduce the storage requirements of the model without compromising its performance. While these approaches have shown some success in reducing the size of time series models, they often focus on model-centric methods rather than data-centric approaches. Data-centric approaches towards time series model compression include lightweight data compression methods based on data statistics and deviation Vidhi Agrawal. Another method involves using real-world data compressors for time series forecasting, where multiple data compressors are combined into one forecasting method with the automatic selection of the best algorithm for the input data K.S. Chirikhin [2019]. Additionally, a two-level compression model has been proposed that selects a proper compression scheme for each individual point, capturing diverse patterns at a fine granularity Marjai et al. [2021]. Furthermore, a data-centric approach has been proposed for detecting anomalous data points in time series data, achieving 100% performance in correctly identifying anomalies Yanjun Zhou [2021]. Finally, a data compressing apparatus has been developed that performs a polygonal line approximation process on time series data Adrián Gómez-Brandón a [2021]. In line with these existing approaches, our research focuses on developing a data-centric approach towards time series model compression.

Definitions

- **Time Series Data:** Let $\mathcal{D}$ represent the time series dataset, consisting of N data points:
$$\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$$
where x_i represents the ith data point in the time series.
- **Training Dataset:** The training dataset, denoted as $\mathcal{D}_{\text{train}}$, comprises a subset of $\mathcal{D}$ used for model training:
$$\mathcal{D}_{\text{train}} = \{x_1, x_2, \ldots, x_{n_{\text{train}}}\}$$
where n_{train} is the number of data points in the training dataset.
- **Validation Dataset:** The validation dataset, denoted as $\mathcal{D}_{\text{val}}$, is a subset of $\mathcal{D}$ used for model selection and tuning:
$$\mathcal{D}_{\text{val}} = \{x_{n_{\text{train}}+1}, x_{n_{\text{train}}+2}, \ldots, x_{n_{\text{train}}+n_{\text{val}}}\}$$
where n_{val} is the number of data points in the validation dataset.
- **Test Dataset:** The test dataset, denoted as $\mathcal{D}_{\text{test}}$, is employed to assess the performance of the time series forecasting model:
$$\mathcal{D}_{\text{test}} = \{x_{n_{\text{train}}+n_{\text{val}}+1}, x_{n_{\text{train}}+n_{\text{val}}+2}, \ldots, x_N\}$$
where n_{test} is the number of data points in the test dataset.
- **Time Series Forecasting Model:** A time series forecasting model, M, is a function that maps a sequence of past data points to predict future data points:
$$M : [x_1, x_2, \ldots, x_t] \rightarrow \hat{x}_{t+1}$$
where $\hat{x}_{t+1}$ represents the predicted data point at time $t + 1$.

Mathematically Represented Problem

The problem at hand can be mathematically formulated as follows: Given a time series dataset $\mathcal{D}$, we aim to find a representative data segment S^* from $\mathcal{D}$ that minimizes the Root Mean Square Error (RMSE) in time series forecasting. Formally, our objective is to find S^* such that:

$$S^* = \arg\min_{S \subset \mathcal{D}} \text{RMSE}(M(\mathcal{D}_{\text{train},S}), \mathcal{D}_{\text{val}})$$

where $\mathcal{D}_{\text{train},S}$ represents the training dataset created using the selected segment S, and $\text{RMSE}(\cdot)$ denotes the Root Mean Square Error between the predicted values using M and the actual values in the validation dataset $\mathcal{D}_{\text{val}}$.

Our objective is to find the segment S^* that optimally balances the trade-off between data reduction and forecasting accuracy, thus improving the efficiency of time series forecasting models while preserving their predictive capabilities.

Data Split and Segmentation

To initiate our approach, we employ a structured data splitting and segmentation strategy. The time series dataset $\mathcal{D}$ is divided into three distinct subsets: the training dataset ($\mathcal{D}_{\text{train}}$), the validation dataset ($\mathcal{D}_{\text{val}}$), and the test dataset ($\mathcal{D}_{\text{test}}$). This division is conducted with a predetermined proportion of 60%, 20%, and 20%, respectively. Within the training dataset, we further partition the data based on a parameter selected by the user. In our experimentation, we segment on a weekly basis, creating distinct segments labeled as week1, week2, week3, and so forth. The validation dataset assumes a crucial role in the process of identifying candidate segments, ultimately leading us to select the representative data segment. The test dataset, reserved for model evaluation, allows us to assess the performance of the selected representative data segment.

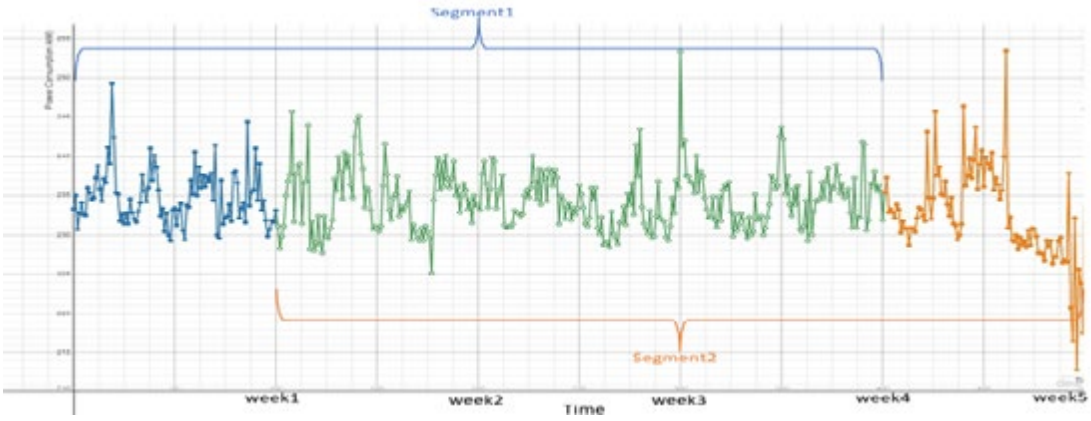

Figure 1: Pictorial Representation of Segments Derived for Training in The Iterations

Iterative Model Training and Selection

Our methodology is characterized by an iterative model training and selection process, which is instrumental in narrowing down the choice of the representative data segment. This process encompasses the following key steps:

First Iteration: In the initial iteration, as illustrated in Figure 1 4.2, we initiate the process by creating two data segments from the training dataset. The first segment is formed by excluding the data from the last week, and the second segment excludes the data from the first week. We proceed to train two distinct time series forecasting models using these two segments. Subsequently, we compare the Root Mean Square Error (RMSE) observed on the validation dataset for these models. The segment associated with the model exhibiting the lower RMSE is selected for further iterations.

Subsequent Iterations: The iterative cycle continues, with the segment selected from the previous iteration serving as the foundation for the subsequent one. In each iteration, two new segments are created, one excluding the data from the last week and the other excluding the data from the first week. Once more, two models are trained using these segments, and the RMSE on the validation dataset is compared. The segment corresponding to the model with the lower RMSE is chosen to advance to the next iteration.

This iterative process persists until a minimum of two weeks' worth of data is available for segment creation. Subsequently, the data segment from the iterations that exhibits the lowest error rate on the validation data is identified as the representative data segment essential for training an effective time series forecasting model.

Innovative Data-Centric Approach

The approach delineated above is both novel and data-centric in its nature. It seeks to identify the representative data segment that optimally balances the trade-off between data reduction and forecasting accuracy, ultimately improving the efficiency of time series forecasting models while preserving their predictive capabilities. Our approach offers a pioneering solution to the challenge of selecting a representative data segment within a time-series dataset, fostering a more efficient and insightful approach to time series forecasting.

Experiments

In our pursuit to validate the effectiveness of the proposed data-centric approach in addressing the challenges of model compression for time series forecasting, we conducted comprehensive experiments. These experiments utilized power utilization time series data collected from 9 servers over 18 weeks, spanning from November 1, 2021, to March 30, 2022. The dataset records the average input power consumed by each server at hourly intervals.

Experimental Setup

For our experiments, we employed Meta's Prophet algorithm to train time series forecasting models. To assess the efficacy of our proposed approach, we conducted two comparative studies:

Comparison with Standard 80-20 Split

The first study involved comparing the performance of time series models trained using the proposed approach with a standard 80-20 data split. The Root Mean Square Error (RMSE) was used as the evaluation metric. The results of this comparison are summarized in Table 11.

System ID	RMSE (Standard 80-20)	RMSE (Representative Data)	Reduction in Error Rate (%)
server0	29.37	29.31	0.2%
server1	21.10	19.84	5%
server2	40.52	22.10	45.45%
server3	18.77	18.42	1.86%
server4	21.29	26.40	-24%
server5	16.79	18.15	-8.1%
server6	43.20	29.91	30.7%
server7	56.48	41.74	26.09%
server8	27.37	41.33	-51%

Table 1: Comparing Error Rates of Models Trained Using the Standard 80-20 Split Approach Against the Proposed Approach For 9 Different Servers.

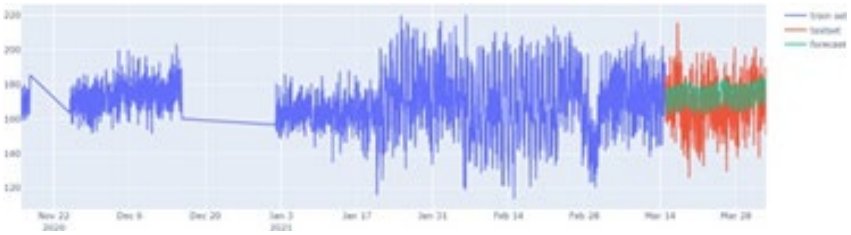

Figure 2: Plot of Trainset, Test set And Forecasted Data Using Model Trained Using Standard 80-20 Split

Comparison with Models Trained with 60% Data

In the second study, we compared the performance of models trained with the proposed approach against models trained using only 60% of the data. Again, RMSE served as the evaluation metric, and the results are captured in Table 22.

Results

Table 11 presents the comparison of error rates between models trained using the standard 80-20 split approach and the proposed data-centric approach for the 9 different servers. Notably, 6 out of 9 time-series models trained using the representative data segment exhibited lower error rates compared to the models trained using the standard 80-20 split approach. This observation underscores the effectiveness of our proposed approach in enhancing forecasting accuracy. In Table 22, we compare the error rates of models trained using 60% of the data against those trained using the proposed data-centric approach for the same set of 9 servers. Encouragingly, 7 out of 9 time-series models trained with the representative data segment demonstrated lower error rates compared to models trained with only 60% of the data. This reaffirms the effectiveness of our approach, even in comparison to the utilization of a larger data portion.

Furthermore, we investigated the reduction in model sizes when trained with the representative data segment. Table 4 details the comparison of time series model sizes for the 9 servers when trained using the standard 80-20 split approach against our proposed approach. Strikingly, there was a total reduction of 56.31% in space consumption, demonstrating the significant efficiency gains associated with our data-centric approach. In summary, our experiments reveal that the proposed data-centric approach leads to improvements in time series forecasting accuracy, with substantial reductions in model sizes. This approach not only enhances computational

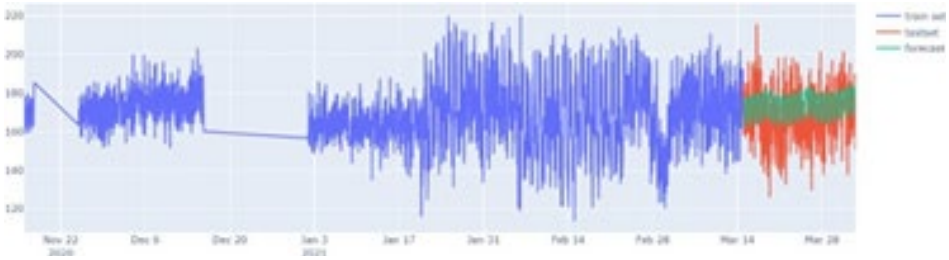

Figure 3: Plot of Representative Data Segment, Test Set and Forecasted Data Using Model Trained Using Proposed Approach.

System ID	RMSE (60% Data)	RMSE (Representative Data)	Reduction in Error Rate (%)
server0	64.37	29.31	54.46%
server1	32.39	19.84	38.74%
server2	41.22	22.10	46.38%
server3	26.86	18.42	31.42%
server4	38.31	26.40	31.08%
server5	22.35	18.15	18.8%
server6	112.07	29.91	73.31%
server7	37.91	41.74	-10.10%
server8	37.05	41.33	-11.55%

Table 2: Comparing Error Rates of Models Trained Using 60% Of the Data Against The Proposed Approach For 9 Different Servers.

System ID	Representative Data Segment Date Range	Representative Data Segment Size (in weeks)	Reduction in Error Rate (%)
server0	14th Dec 2020 – 21st Feb 2021	9	54.46%
server1	14th Dec 2020 – 7th Feb 2021	7	38.74%
server2	4th Jan 2021 – 31st Jan 2021	4	46.38%
server3	25th Nov 2020 – 7th Feb 2021	10	31.42%
server4	1st Feb 2021 – 7th Feb 2021	1	31.08%
server5	2nd Jan 2021 – 21st Feb 2021	8	18.8%
server6	7th Dec 2020 – 14th Feb 2021	9	73.31%
server7	1st Feb 2021 – 7th Feb 2021	1	-10.10%
server8	25th Nov 2020 – 21st Feb 2021	12	-11.55%

Table 3: Data Range and Size of Representative Data Segment for Each Server.

efficiency but also contributes to the interpretability and scalability of time series forecasting models. The results presented here validate the transformative potential of our approach in the realm of time series analysis.

Conclusion

In this paper, we introduced a novel data-centric approach designed to address the challenges of model compression in time series analysis. Our methodology, grounded in the idea that not all data points in a time-series dataset are equally informative, advocated for the identification of a representative data segment as the foundation for training effective time series models. Through a comprehensive series of experiments, we explored the effectiveness of this approach. Our results clearly demonstrate the significant advantages of our data-centric approach. When compared to the standard 80-20 split, our approach consistently exhibited lower root mean square error (RMSE) on the test set for the majority of the servers, illustrating its superior predictive performance. Moreover, the reduction in error rate was most pronounced when compared to models trained with only 60% of the data, highlighting the efficiency of our proposed approach.

Not only did our approach improve predictive accuracy, but it also addressed the issue of model size. The models trained with the representative data segment were notably more compact, resulting in a 56.31% reduction in model size on average. This reduction is of paramount importance for resource optimization, particularly in scenarios where storage space and computational resources are at a premium. Furthermore, our data-centric approach also aligns with the need for model interpretability. By focusing on the most informative data segments, we simplified model complexity, making it easier to understand the underlying patterns and relationships within the data. This improvement in interpretability has practical implications for decision-making and forecasting in various fields. In summary, our research offers a pioneering method to address the challenges of data volume and model complexity in time series analysis. By focusing on a representative data segment, we have demonstrated the potential for more efficient and insightful modeling, both in terms of predictive performance and resource optimization. We believe that our data-centric approach opens up new possibilities for enhancing computational efficiency and interpretability in the realm of time series analysis, making it a valuable tool for a wide range of applications across different domains.

System ID	Size of Model (Standard 80-20, Bytes)	Size of Model (Representative Data, Bytes)	Reduction in Model Size (%)
server0	454,615	256,557	43.56%
server1	454,247	189,232	58.34%
server2	454,201	138,036	69.60%
server3	454,309	277,887	38.83%
server4	454,210	37,988	91.63%
server5	454,276	248,255	45.35%
server6	454,137	254,901	43.87%
server7	454,222	38,018	91.63%
server8	454,258	345,105	24.02%
Total	4,088,475	1,785,979	56.31%

Table 4: Comparison of Time Series Model Sizes Of 9 Different Servers Trained Using the Standard 80-20 Split Approach Against the Proposed Approach.

This paper encourages further exploration of data-centric approaches and their application in addressing the computational and interpretative challenges of time series analysis. As the volume of time series data continues to grow, these innovative methodologies are crucial for staying ahead in the realm of data-driven decision-making and forecasting.

References

1. Rasheed, F., Alshalalfa, M., & Alhajj, R. (2010). Efficient periodicity mining in time series databases using suffix trees. IEEE Transactions on Knowledge and Data Engineering, 23(1), 79-94.
2. Almeida, A., Brás, S., Sargento, S., & Pinto, F. C. (2023). Time series big data: a survey on data stream frameworks, analysis and algorithms. Journal of Big Data, 10(1), 83.
3. Shu, W., Wang, W., & Wang, Y. (2014). A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. EURASIP Journal on Wireless Communications and Networking, 2014(1), 1-9.
4. Yu, D., & Xiao, J. (2022). Multilevel Information Granule Construction Model Based on Large Data Environment and Its Application in Time Series. Mobile Information Systems, 2022.
5. Michael H. Zhu and Suyog Gupta. pruning for model compression - arxiv.org.
6. Yin, X. X., Miao, Y., & Zhang, Y. (2022). Time series-based data explorer and stream analysis for anomaly prediction. Wireless Communications and Mobile Computing, 2022.
7. Bhattarai, B. P., Paudyal, S., Luo, Y., Mohanpurkar, M., Cheung, K., Tonkoski, R., ... & Zhang, X. (2019). Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions. IET Smart Grid, 2(2), 141-154.
8. Ryabko, D., & Mary, J. (2012). Reducing statistical time-series problems to binary classification. Advances in Neural Information Processing Systems, 25.
9. Barez, F., Bilokon, P., & Xiong, R. (2023). Benchmarking Specialized Databases for High-frequency Data. arXiv preprint arXiv:2301.12561.
10. Pietron, M., & Wielgosz, M. (2020). Retrain or not retrain?-Efficient pruning methods of deep CNN networks. In Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III 20 (pp. 452-463). Springer International Publishing.
11. Dhananjoy Dey Vidhi Agrawal, Gajraj Kuldeep. Near lossless time series data compression methods using statistics.
12. Chirikhin, K. S., & Ryabko, B. Y. (2019). Application of data compression techniques to time series forecasting. arXiv preprint arXiv:1904.03825.
13. Marjai, P., Lehotay-Kéry, P., & Kiss, A. (2021). The use of template miners and encryption in log message compression. Computers, 10(7), 83.
14. Zhou, Y., Ren, H., Li, Z., Wu, N., & Al-Ahmari, A. M. (2021). Anomaly detection via a combination model in time series data. Applied Intelligence, 51, 4874-4887.
15. Gómez-Brandón, A., Paramá, J. R., Villalobos, K., Illarramendi, A., & Brisaboa, N. R. (2021). Lossless compression of industrial time series with direct access. Computers in Industry, 132, 103503.