Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

GENERATING TCN MODELS FROM PARALLEL DEVS MODELS: SEMICONDUCTOR MANUFACTURING SYSTEMS

Vamsi Krishna Pendyala[1], Hessam S. Sarjoughian[1], and Edward J. Yellig[2]

[1]Arizona Center for Integrative Modeling & Simulation, Arizona State University, Tempe, AZ, USA
[2]Intel Corporation, Chandler, AZ, USA

ABSTRACT

Machine learning models have the potential to augment the simulation of discrete-event dynamical systems, which is of considerable interest. Such models should serve the central purpose of capturing the temporal dynamics of event-based systems. In this paper, we use simulations of Parallel DEVS (PDEVS) models of a benchmark semiconductor fabrication manufacturing system to generate ARIMA, RNN, LSTM, and TCN models of the same. Single/multi-stage manufacturing statistical and deep learning models are developed and evaluated for different experimental scenarios. We generate Temporal Convolutional Neural (TCN) network models and evaluate their uni/multivariate throughput and turnaround time series by varying wafer lot configurations and sizes. The results show the predicted time series generated by TCN models can approximate the accuracies of simulated PDEVS models while achieving many-fold execution speedup.

1 INTRODUCTION

Many systems are undergoing significant changes in their operations due to having higher complexities and scales. Considering smart semiconductor manufacturing, substantial resources and time are needed to build and operate due to inevitable changes in the supply chain of materials, advances in tools, and variability in product types and demand quantities. Deductive simulatable models are widely used to gain insight into what manufacturing systems should do and how to operate at individual and aggregate levels. Recently, Machine Learning (ML) has been gaining attention for developing models for purposes ranging from understanding and discovering their known and hidden dynamics to developing requirements and designs for factories of the future (Cimino et al. 2019; Griffiths and Ooi 2018; Chien et al. 2023; von Rueden et al. 2021).

In earlier work, Parallel DEVS (PDEVS) single-stage and cascade models based on a benchmark MiniFab Intel factory (Spier and Kempf 1995) are developed (Sarjoughian et al. 2023). We formulated a suite of simulation experiments to process all received wafer lots with varying configurations and sizes. Collected simulation data sets are processed and used to create a set of regression algorithms, followed by evaluating how well each can predict specific variables such as factory throughput and turnaround time alongside generic measures such as Mean Absolute Error (*MAE*). Factory dynamics predicted using Automatic Relevance Determination (ARD), Decision Tree (DT), and K-Nearest Neighbors (KNN) algorithms show negligible differences compared to the throughput predictions obtained from the PDEVS models. If such accurate data-based models can be created and reused, they execute much faster compared to their knowledge-based models (Sarjoughian et al. 2023). This kind of regression model, however, does not lend itself to simulating the behavior across chronological time steps. Deep learning models such as Temporal Convolution Networks (TCN) (Bai et al. 2018) may. Unlike regression models, deep learning models can produce time series akin to time trajectories in simulation models. The accuracy of such time series (e.g., factory throughput) can be measured/evaluated at regular time intervals.

The research in this paper aims to identify and harness the capabilities of Deep Learning models to gain temporal behavior of semiconductor fabrication manufacturing processes. Building upon established

979-8-3315-3420-2/24/$31.00 ©2024

research that underscores the essential use of Parallel Discrete Event Simulation (PDEVS) models (Chow and Zeigler 1994) for manufacturing systems, our focus is on using Machine Learning models for temporal analysis. The aim of creating these models is to encapsulate the essence of real-life scenarios and to harvest data, providing useful insights without the need for actual manufacturing systems. The data, a valuable byproduct of simulation, serves a dual purpose — it becomes a resource for refining existing models and developing others to find new insights into product flows and gain execution speedup.

This paper shows the use of the Recurrent Neural Network (RNN) (Rumelhart et al. 1986), Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), and TCN Deep Learning and Autoregressive Integrated Moving Average (ARIMA) (Box and Jenkins 1970) models for single/multi-stage MiniFab models using the simulated data sets obtained from PDEVS models. We compare the predicted accuracy of throughput and turnaround time of the ARIMA, LSTM, RNN, and TCN models to those of the PDEVS models. The TCN models are used to predict the throughput and turnaround times of single stages of multi-stage cascade models from the preceding and current stages. Accurate TCN models, under a wide range of lot sizes and configurations, can be used as surrogates of the PDEVS models. This study shows the TCN model outperforms ARIMA, LSTM, and RNN models in terms of standard accuracy metrics and throughput and turnaround time trajectory predictions compared with those from the PDEVS simulations.

2 RELATED WORKS

Knowledge-based Discrete Event Simulation (DES) is commonly used in the analysis and enhancement of manufacturing processes (e.g., Xie and Allen 2015). The intricacies inherent in manufacturing systems, particularly in domains like semiconductor manufacturing, pose challenges that can be effectively addressed through the application of DES (Ehm and Ponsignon 2012). This methodology aids in the analysis of critical parameters such as throughput, turnaround time, utilization, and overall efficiency of realistic systems (Kampa et al. 2017).

A practical application where data collection from each machine in a truck manufacturing line, including metrics like DownTime (DT), led to a notable improvement of 12% in the block line (Ingemansson and Oscarsson 2005). This simulated data, in turn, can become a valuable resource for the development of predictive models, including machine learning models, enabling a comprehensive analysis of the simulated model. Synthetic manufacturing datasets for machine learning can be generated through DES models (Chan et al. 2022). This work describes the critical role of data in the development of machine learning models and illuminates the inherent limitations of relying solely on real-world production systems for data generation. From a broader perspective, the integration of machine learning into large-scale manufacturing industries is made more attainable and efficient through the utilization of DES for data generation (Mazzei and Ramjattan 2022).

Machine/Deep Learning (ML/DL) based predictive models are effective for real-world systems that are inherently complex. They can be used to develop predictive models based on many parameters and volumes of historical, temporal, or spatial datasets (Sarker 2021). Specifically, predictive models have been extensively used in the domain of manufacturing (Goldman et al. 2021; Lingitz et al. 2018; Subramaniyan et al. 2018; Qiao and Wang 2021; Fronckowiak et al. 1996).

The application of discrete event simulation along with predictive modeling is discussed in (Bartos et al. 2017), where an ARIMA time series model is used to provide forecast inputs to a DES model used in a hospital system service. However, due to the limitations of ARIMA time series models in dealing with large and nonlinear datasets, they can be replaced with ML/DL models and integrated with a DES model for effective results. The potential of integrating Machine Learning models (ML) for job-shop production is described in (Lang et al. 2020). The integration of ML/DL models or the use of ML/DL models to predict simulation output can provide new insights and, in certain cases, help achieve output at a much faster rate than DES models alone (Atalan et al. 2022; Sarjoughian et al. 2023). The research mentioned above involves either integrating or implementing predictive models along with DES models to improve a system's efficiency and understand its hidden dynamics.

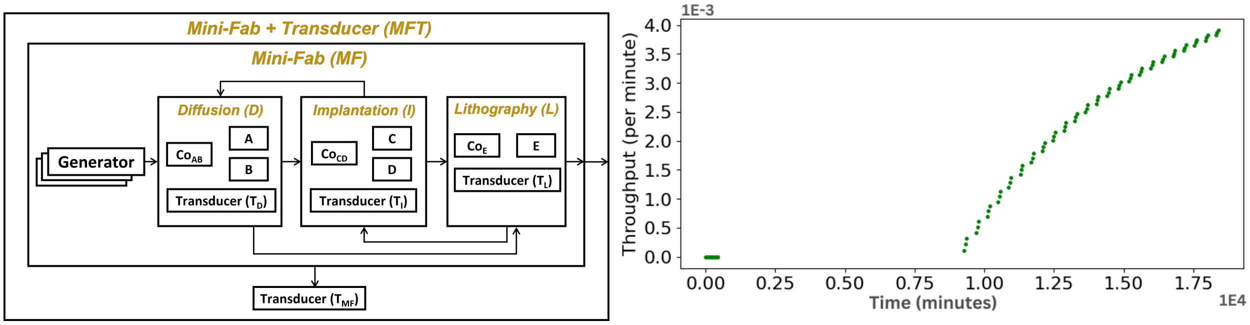
Machine learning methods (Choi and Kim 2002; Saadawi et al. 2016) and multi-level abstraction (Seok et al. 2020) have been proposed and developed for simulating discrete systems, including semiconductor manufacturing. In our earlier work, predictive atemporal regression ML models were developed using the PDEVS of the semiconductor fabrication manufacturing briefly described below (Sarjoughian et al. 2023). A study focused on the production flow analysis of the benchmark semiconductor fabrication manufacturing system implemented in Anylogic using several atemporal regression models for a select lot configuration choices (Singgih 2021).

3 EXEMPLAR SEMICONDUCTOR MANUFACTURING FACTORY MODEL

Developing machine learning models using data from actual semiconductor manufacturing factories is highly resource-intensive and subject to proprietary restrictions. The simulations of the semiconductor fabrication manufacturing PDEVS models described below, however, are not subject to such limitations. The DEVS-Suite simulator enables flexible model development, systematic design of experiments, efficient simulation, and automation for data collection and database storage (ACIMS. 2023).

Here, we describe single-stage and multi-stage semiconductor factory models developed using the Parallel Discrete Event Specification (PDEVS) formalism & DEVS-Suite simulator, and the data derived from their simulations (Chow and Zeigler 1994; ACIMS. 2023). PDEVS factory models have been developed based on a single-stage model of an Intel manufacturing factory (Spier and Kempf 1995). Each stage has Diffusion, Implantation, and Lithography coupled components and machines A, B, C, D, and E atomic components. Machines A and B are identical as are machines C and D. Each machine is managed by a coordinator atomic component responsible for dispatching wafers. The assignment of wafers to machines occurs instantaneously, consuming zero logical time during simulation. Each single-stage factory processes wafer lots designated as Product a (Pa), Product b (Pb), and Test wafer (Tw). These lots must be assembled into batches of three before proceeding chronologically through a six-step process across Diffusion, Implantation, and Lithography, with steps 1 and 5 for machines A and B, steps 2 and 4 for machines C and D, and steps 3 and 6 for machine E. Batching rules dictate that only one Tw lot can be included in a batch at most, and while Pa and Pb lots can be mixed in step 1, they cannot be combined in step 5. Each machine operates in four consecutive phases: loading, processing, unloading, and transporting of wafer lots, with assigned periods subject to random uncertainty (e.g., 10%). Additionally, transducers are deployed to collect data from machines and coordinators within each part of a stage. These transducers do not have any side effects on the operations of the machines and their relationships. The time unit to capture discrete events in the simulation is measured in terms of minutes. An illustration of the workflow of a single-stage factory is depicted in Figure 1 part (a). These single-stage models are coupled together to form a multi-stage PDEVS model with n stages and each stage with a transducer to record values such as throughput & turnaround time. For experiments performed in this paper involving multi-stage MiniFab, we have used an 8-stage PDEVS MiniFab model.

After running a simulation, we use throughput and turnaround time collected from the machines for stages $\mathscr{S}_i \in \{\mathscr{S}_1, \cdots \mathscr{S}_8\}$. As mentioned above, the Pa, Pb, and Tw (lot configuration) lots must be ordered to be processed in the simulation. Sample values of throughput collected over time during the simulation for a single stage are as depicted in Figure 1 part (b). We have devised 93 different lot configurations with different lot sizes (e.g., Pa=10, Pb=90, Tw=20) and collected data from the simulations for all the configurations. The stored data sets serve our primary objective to analyze the overall throughput and turnaround time from each stage and develop predictive models for the same. To create a model capable of predicting throughput and turnaround time for the next time unit (next minute), it's imperative to analyze the data as time series. They provide deeper insights into the data compared to general regression models, which predict throughput at the end of each simulation scenario rather than capturing time trajectories of throughput and turnaround time.



(a) Component-based PDEVS model.

(b) (Pa=54, Pb=0, Tw=18) lot configuration.

Figure 1: Single-stage MiniFab semiconductor manufacturing factory model.

4 THROUGHPUT AND TURNAROUND TIME FORECASTING

Consider a cascade semiconductor manufacturing model with n-stages. Let each stage be denoted as $\mathscr{S}_i \in \{\mathscr{S}_1, \cdots, \mathscr{S}_n\}$. Each stage has a throughput denoted by $\mathscr{TH}^i \in \{\mathscr{TH}^1, \cdots, \mathscr{TH}^n\}$ and a turnaround time denoted by $\mathscr{TA}^i \in \{\mathscr{TA}^1, \cdots, \mathscr{TA}^n\}$. To predict throughput and turnaround time at a given time, we need the throughput and turnaround time at fixed time intervals. However, the data obtained from the simulation encompasses varying time intervals dictated by the time to the next event of each machine in every stage. Consequently, the data sets cannot be directly utilized for time series analysis, necessitating a data preprocessing step to prepare the data for creating predictive models discussed further in Section 4.

4.1 Data Preprocessing

The data described in Section 3 must be preprocessed for further timeseries analysis. The simulated input and output are discrete event trajectories. Each trajectory has a continuous time base with a finite number of values for any given finite time interval along with unknown values at a given time instance (as seen in Figure 1 part (b)). The time base of the discrete event data trajectory should be converted into a discrete time base for time series analysis. Transforming discrete event data into discrete-time data involves assigning a finite discrete time interval to each discrete event data. For a simulation with a continuous time base, we can convert any discrete event data trajectory into time series data with a desirable time granularity (e.g., 1 minute).

For our experiments, we have used 1 minute as the time interval for the discrete-time base. Also, for the discrete-time instances for which no data is available in the discrete event trajectory, we used forward filling to add a value for every time instance of a time series (e.g., every null value associated with a time instance is replaced with the value preceding it). Forward filling of null data is performed since variable trajectories of every atomic DEVS model are piece-wise constant and output events are defined in terms of the model's state. The throughput and turnaround time after data preprocessing for an 8-stage cascade factory with a lot size of 120 (Pa=10, Pb=90, Tw=20) are shown in Figure 2.

Now, we define a formalism for our throughput and turnaround time values as, for a given time $t \in \mathbb{N}$, the throughput and turnaround time can be denoted as $\mathscr{TH}_t^i$ and $\mathscr{TA}_t^i$ respectively for a given stage i and a time t. We can analyze and pre-process the throughput and turnaround time values of different configurations of Pa, Pb, and Tw of the MiniFab similarly.

4.2 Timeseries Models and Single-stage Analysis

For throughput and turnaround time forecasting, we consider a k-step delay prediction, which can be described as - to predict a value (throughput or turnaround time) at time t, a time series model would use previous values in time from $t-1$ to $t-k$. More formally, the problem statement can be written as – *given*

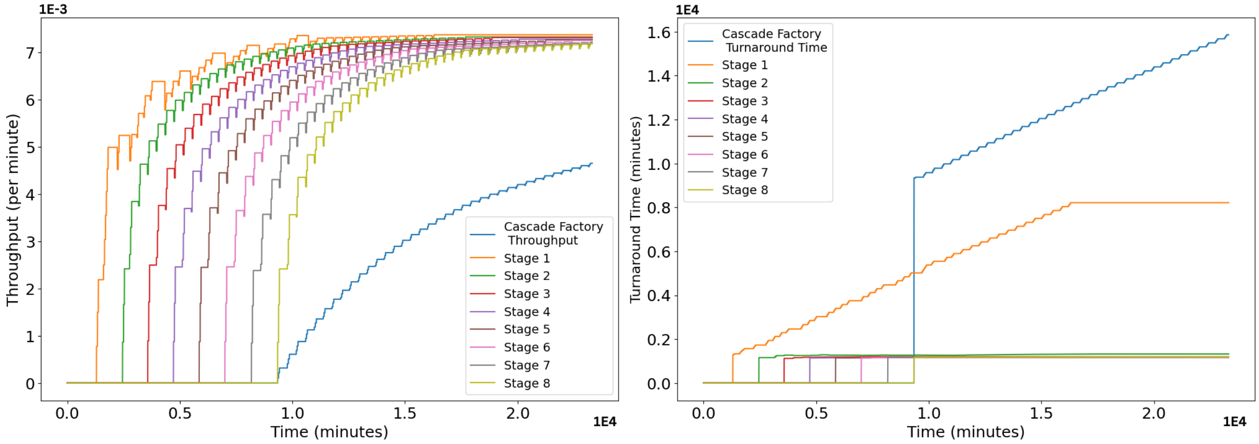


Figure 2: Preprocessed $\mathscr{TH}$ and $\mathscr{TA}$ for stages $\mathscr{S}_1, \cdots, \mathscr{S}_8$ cascade factory (Pa=10, Pb=90, Tw=20).

Table 1: 8-stage MiniFab timeseries model comparison.

	Throughput predictions				**Turnaround time predictions**			
Model	*MSE*	R^2 *Score*	*MFE*	MAPE	*MSE*	R^2 *Score*	*MFE*	MAPE
ARIMA	3.00E-07	-6.006	5.07E-04	8.13%	213068	-3.843	411	15.66%
RNN	1.35E-08	0.681	1.04E-04	1.67%	175	0.995	11	0.66%
LSTM	9.73E-09	0.771	-9.35E-05	1.53%	158	0.996	-7	0.44%
TCN	2.29E-09	0.946	-4.56E-06	0.51%	34	0.999	1	0.08%

past data for $Throughput^i_{t-k} = \{\mathscr{TH}^i_1, \cdots, \mathscr{TH}^i_{t-k}\}$*, and* $TurnaroundTime^i_{t-k} = \{\mathscr{TA}^i_1, \cdots, \mathscr{TA}^i_{t-k}\}$ *the aim is to predict throughput* $\mathscr{TH}^i_t$ *and turnaround time* $\mathscr{TA}^i_t$ *at the current time* t *for the stage* i *based on it's past values.* This can be achieved using multiple time series models such as ARIMA, LSTM, RNN, and TCN to predict the values of throughput and turnaround time based on their past values. We set the output for each of these models to be $\mathscr{TH}^i_t$ or $\mathscr{TA}^i_t$ at time t for stage i and the input can be the values of throughput and turnaround time k time-steps before the current time t i.e., $\mathscr{TH}^i_{t-k}$ or $\mathscr{TA}^i_{t-k}$. The single-stage analysis comprises an analysis of throughput and turnaround time for a particular stage to evaluate the accuracy and predictability of various time series models. The process involves developing a model to predict throughput and turnaround time as mentioned in Section 3. To analyze the performance of each of the time series models, we train and test baseline models of each of the models. The dataset used for the analysis of each of the time series models consists of 5,000-time steps of throughput and turnaround time (66% training & 34% testing) of the MiniFab cascade factory. The results for training and testing each of the models are as in Table 1 where we have considered Means Square Errors (MSE), R^2 Scores, Mean Forecast Errors (MFE), and Mean Absolute Percentage Errors (MAPE) as performance evaluation metrics of our forecasting models (Mehdiyev et al. 2016). The plots of individual models for throughput and turnaround time forecasting are depicted in Figure 3 respectively.

The results show us that the TCN model gives us better results, is also faster to train, and is less sensitive towards sequence length compared to other models. Since it is shown that the TCN model shows higher potential for our use case, the next set of experiments comprises analyzing the performance of a TCN model for predicting throughput and turnaround time for a MiniFab factory model based on simulation data using the TCN model. Now, we have considered two different profiles of throughput and turnaround time with the same lot size but different configurations for training, validation, and testing purposes. For tuning our hyperparameters (e.g., no of hidden layers, activation function, loss function for a neural network) during the validation process, we have used random-search strategy (James and Yoshua 2012) to develop

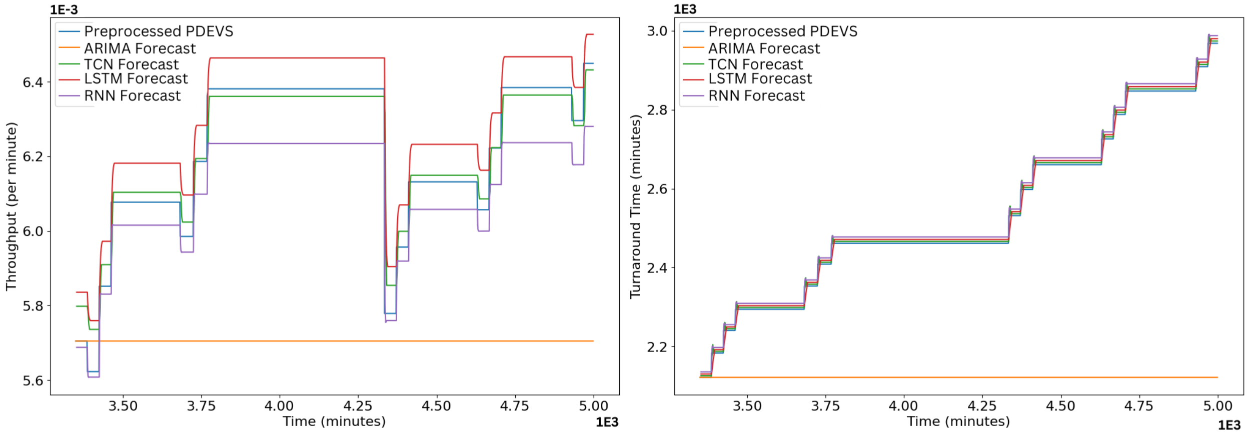


Figure 3: Model comparison.

a model with better prediction trajectory and accuracy. Based on our formalism, we form the algorithm to develop a time series model as mentioned in Algorithm 4.2 (Timeseries Modeling Algorithm) where our result metrics to evaluate a predictive model constitutes of Mean Square Errors, R^2 Scores, Mean Forecast Errors, and the throughput & turnaround time trajectories. The configuration used for dataset A is – Pa=10, Pb=90, Tw=20, and for dataset B is Pa=9, Pb=96, Tw=15, and the training and validation split used for S^A_{train} and S^A_{valid} was 66% and 34% respectively. Once the time series models are developed, they are further analyzed to derive insights related to the MiniFab model.

Algorithm 1 Timeseries Modeling Algorithm.

```
Require: Simulation datasets $S^A_t$ and $S^B_t$ of throughput $Throughput^i_t = \{\mathscr{TH}^i_1, \cdots, \mathscr{TH}^i_t\}$ and turnaround time $TurnaroundTime^i_t = \{\mathscr{TA}^i_1, \cdots, \mathscr{TA}^i_t\}$ in a time period $t$ for a stage $i$ for two different lot configurations $A$ & $B$
Split data from dataset $S^A_t$ into training and validation as $S^A_{train}$ and $S^A_{valid}$
Consider $n$ different hyperparameter combinations $\leftarrow \{hp_1, hp_2, \cdots, hp_n\}$
for $i = 1, 2, \cdots, n$ do
    Consider a time series model $m_i$ with hyperparameters $hp_i$
    Consider a look-back window $k$
    while Training Phase do
        Train model $m_i$ with $(S^A_{t-k}, \cdots, S^A_{t-1}) \in S^A_{train}$ as input and $S^A_{t-1} \in S^A_{train}$ as output
    end while
    while Validation Phase do
        Predict $S^A_t \in S^A_{valid}$ based on input $(S^A_{t-k}, \cdots, S^A_{t-1}) \in S^A_{valid}$ using model $m_i$
        Compute results of predictions for model $m_i$
    end while
    Store results for $m_i$
    Store model $m_i$
end for
Select model $m_k \in \{m_1, m_2, \cdots, m_n\}$ with the best results
Test selected model $m_k$ on dataset $S^B_t$
Evaluate results of model $m_k$ for testing on dataset $S^B_t$
```

4.3 Lot Size Impact

As we possess a dataset derived from the model's simulation encompassing various lot configurations (Section 3), we seek to determine an optimal lot size for training a time series model for achieving better accuracy and learn about the impact of lot size on overall predictive model accuracy. As we can see the throughput profile of each configuration as shown in Figure 1 and Figure 2 have similar trends but different values at a granular level, hence we wish to find an optimal lot size to develop a robust time series model. This optimal lot size will serve as the basis for constructing a predictive model for the throughput/turnaround time time series prediction of all other configurations. To achieve this, a Temporal Convolutional Neural Network (TCN) model was employed with specific hyperparameter configurations. The model was trained on three distinct categories of lot sizes: small, medium, and large. The 'small' category featured a total lot size of 60 (consisting of Pa=15, Pb=36, Tw=9), 'medium' had a total lot size of 120 (with Pa=10, Pb=90, Tw=20), and 'large' comprised a total lot size of 192 (comprising Pa=150, Pb=24, Tw=18). After training the TCN model on these three designated configurations, comprehensive tests were carried out across all 93 configurations, and results are elaborated in the subsequent Subsection 5.1. The results of this experiment would provide us with an insight into training a time series on a particular lot for achieving a highly accurate model.

4.4 Multivariate Timeseries

In real-life semiconductor manufacturing systems or their simulation counterparts, changes in one part affect the whole system. As shown in Figure 1, the components in the single-stage factory have feedforward and feedback. For the cascade PDEVS models, the throughput and turnaround time of all upstream stages affect all downstream stages either directly or indirectly. Multivariate analysis helps us analyze various combinations of throughput/turnaround time from different stages. Ideally, the time series model developed to predict the throughput and turnaround time of a particular stage should account for all changes to downstream stages that are caused by their immediate upstream stages. We undertake this experiment to examine how throughput and turnaround time of two or three stages can be used to predict these values for a particular stage, in contrast to a univariate approach, where only one time series data is recorded at each instance. To facilitate multivariate analysis, we input multidimensional data from different stages of the MiniFab factory into the TCN model for both training and testing purposes. Before training our TCN model, we implement z-score standardization (aka normalization) on the data to achieve faster computation and better accuracy (Hastie et al. 2001) along with data preprocessing as described in Subsection 4.1. Multivariate time series models are developed in which we provide the throughput and turnaround times values to predict the stages classified into two objectives:

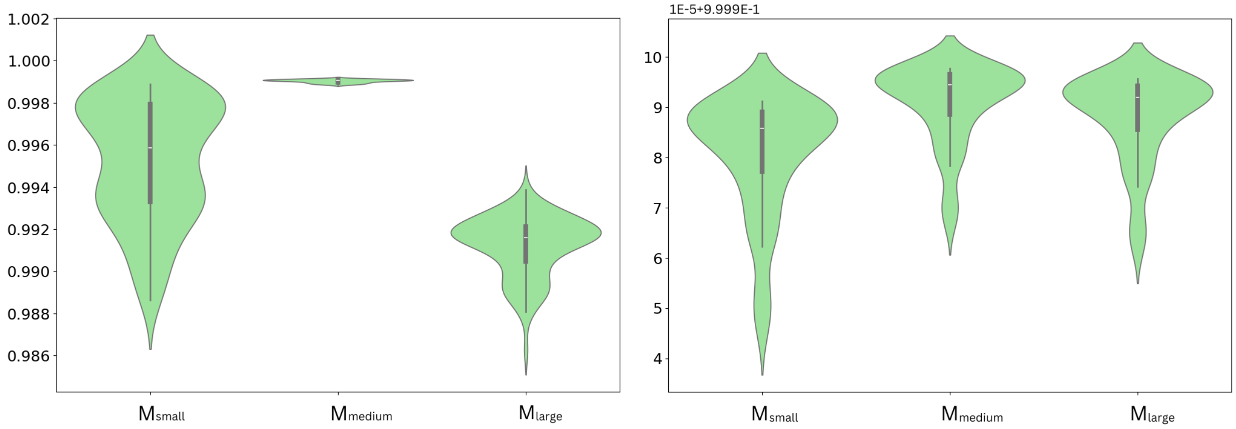
1. Predicting throughput/turnaround time values of a current stage based on values from its preceding stage and the current stage: $\mathscr{T}\mathscr{H}$ and $\mathscr{T}\mathscr{A}$ predictions for $\mathscr{S}_2$ based on $\mathscr{S}_1$ & $\mathscr{S}_2$.
2. Predicting throughput/turnaround time values of a current stage based only on values from its preceding stages: $\mathscr{T}\mathscr{H}$ and $\mathscr{T}\mathscr{A}$ predictions for $\mathscr{S}_5$ based on $\mathscr{S}_4$, $\mathscr{S}_3$ & $\mathscr{S}_2$.

5 EXPERIMENTS AND EVALUATION

To carry out this evaluation, we developed combinations of throughput and turnaround time values of different stages in the 8-stage factory by dividing the stages into three categories. The categories of the stages are based on the progression of different batches as defined for the cascade MiniFab PDEVS models. The categories are *i*: Early stages (1, 2, & 3), *ii*: Intermediate stages (3, 4, 5, 6, & 7), and *iii*: Terminal stages (6, 7, & 8). These categories are used to form 25 unique combinations (19 for objective 1 & 6 for objective 2). Further details and discussion of the results are provided in Section 5. This analysis can help us understand the scope of interchangeability between DEVS models and machine learning models.

5.1 Lot Size Impact Analysis

We trained three TCN models based on different wafer lot sizes and configurations as mentioned in Subsection 4.3. We have named these models – M_{small} is a model trained on small lot size configuration, M_{medium} is a model trained on medium lot size configuration, and M_{large} is a model trained on large lot size configuration. We evaluate the throughput and turnaround predictions for each of the TCN models in terms of MSE and R^2 scores. We tested each of the models against 93 different lot configurations/sizes and collected their R^2 scores. To visualize the results, we have used violin plots (Hintze and Nelson 1998; Tanious and Manolov 2022) to depict the results shown in Figure 4. A violin plot combines box plots and density curves (see Figure 1 in Hintze and Nelson 1998). The widths of the density curves along the y-axis represent the approximate frequency of changes in data points. The line at the center of the density curves is overlaid with boxes showing the median point and interquartile (IQR) range. The median point shows central tendency, while the IQR (25% to 75% range) and whiskers highlight data spread and variability. The violin plots in Figure 4 show us the distribution of R^2 scores across all 93 configurations.



(a) $\mathscr{TH}$ prediction R^2 scores.

(b) $\mathscr{TA}$ prediction R^2 scores.

Figure 4: Cascade factory lot configuration and size impact.

We can observe that models trained in both small and large lot size configurations gave us poor results compared to models trained with medium lot sizes based on their R^2 score distribution. Moreover, we can see that the kernel density plot distribution of R^2 for models trained on small and large lot size configurations doesn't have a uniform distribution when compared with the medium lot size configuration for throughput prediction. We obtained similar results for MSE values and we get the best performance for the model trained on a medium lot size configuration in terms of both MSE and R^2 score. Also for turnaround time, based on R^2 scores for turnaround time prediction we can see that they follow a similar trend as in the case of throughput prediction where the model trained on medium lot size has better performance than the rest, but in terms of the density plots, turnaround time has similar accuracy distributions as compared to throughput results suggesting that throughput values are more susceptible to lot size impact. Considering the box plots in Figure 4, we can see that, in the case of throughput prediction, a model trained on small lot configuration (M_{small}), has a larger difference between the lower quartile and its upper quartile values compared to other models, suggesting that there is a high variability in accuracy for this model. Even though M_{large} has lesser variability compared to M_{small} for throughput prediction, the median of the box plot for M_{large} is still lower than those of other models. The model trained on medium lot size (M_{medium}) has the least variability and the higher & lower quartile values are closer to its median, suggesting that M_{medium} has a higher precision and based on its median, it also has a higher accuracy. We see a similar trend in the case of turnaround time in terms of accuracy, where the median value of M_{medium} is higher than other

Table 2: Multivariate analysis prediction results.

	Mean Square Error				R^2 Score			
	Minimum	Maximum	Average	Median	Minimum	Maximum	Average	Median
Obj-1	5.66E-19	2.87E-18	1.0752E-18	7.37E-19	0.997	0.999	0.998	0.999
Obj-2	1.22E-18	7.40E-18	2.8817E-18	2.14E-18	0.992	0.998	0.997	0.997

models. However we observe less precision for predicting turnaround time than throughput using M_{medium} based on their box plots, this can be attributed to the fact that there is a higher variability of turnaround time values among each lot configuration than throughput values. We also have higher variability at lower accuracies (long lower whisker in Figure 4 (b)). These observations suggest that accuracy is not directly or inversely proportional to the lot size, rather, there exists an optimal lot size and lot configuration to develop a TCN model that can predict throughput and turnaround time for different lot configurations and lot sizes. The R^2 accuracy for throughput is highest for M_{medium} and lowest for M_{large}. The R^2 accuracy for turnaround time is highest for M_{medium} and lowest for M_{small}.

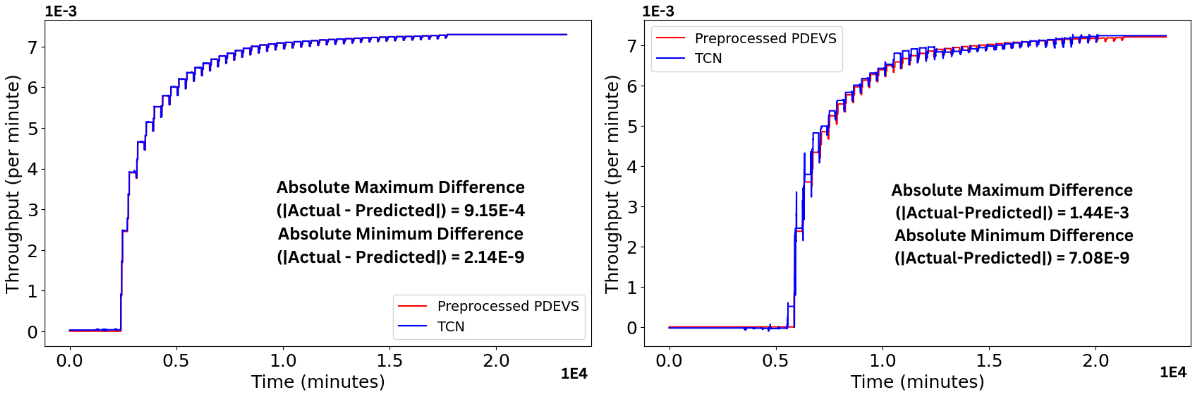
5.2 Multivariate Timeseries Analysis

As discussed in Subsection 4.4 we perform multivariate analysis using the TCN model. We performed multivariate analysis using the throughput (and turnaround time) of connected stages for predicting the throughput and turnaround time of a single stage. Based on the plots in Figure 5, we can see that the TCN model performs well. This kind of analysis helps us develop a model that can predict values and take into account the effect of additional information (i.e., throughput and turnaround time of other stages in our case). Based on the combinations mentioned in Subsection 4.4, we computed 19 scenarios in the case of objective 1 and 6 scenarios in the case of objective 2. The values of MSE for throughput and turnaround time for cascade factory are as shown in Table 2 respectively. The R^2 scores for objectives 1 & 2 can be observed in the Table 2.

The plots indicate that we observe the lowest mean square error for objective 1, where we predict throughput for a stage based on previous stages and the current stage itself. For both MSE and R^2 we observe poor performance for objective 2, where we try to predict the throughput of the current stage based on the previous two stages. This can be attributed to the fact that each lot (Pa, Pb, Tw) is processed successively, and the TCN model predicts the throughput of a stage purely based on its previous 2 stages without the knowledge of the current stage, whereas for objective 1, the model has knowledge of current stage along with its previous stages. Both objectives serve our purpose of incorporating the effects of other stages to predict values for a particular stage. Sample plots to visualize the results of the best and worst performing models are shown in Figure 5. Better accuracy is observed for Figure 5 (a) which corresponds to objective 1 and lower accuracy is observed for Figure 5 (b) which corresponds to objective 2. This is due to the reason mentioned above for different accuracies for objectives 1 and 2. Additionally, higher accuracies were observed with more stages and also towards the end of the simulation. These findings stress considering stage interdependence in semiconductor manufacturing predictive modeling.

5.3 Computing Platform

To carry out our experiments, we have used Python 3.10 as the primary programming language. We used, pandas and numpy libraries for data preprocessing along with SciKit Learn and Tensorflow libraries. All predictive modeling experiments were conducted utilizing a T4 GPU with a clock rate of 2.20 GHz and a 64-bit capacity running on Ubuntu OS. The DEVS-Suite Simulator Version 7.0.0 was used on a GPU with a clock rate of 3.70 GHz and a 64-bit capacity running on a Windows OS for the development and execution of the PDEVS MiniFab models. Experiments were conducted to assess the speed of predicting throughput and turnaround time values using PDEVS simulation and TCN model across small, medium, and large configurations (as described in Section 4.3). We did not consider the training time of the TCN



(a) $\mathscr{TH}$ prediction for $\mathscr{S}_2$ based on $\mathscr{S}_1$ & $\mathscr{S}_2$.

(b) $\mathscr{TH}$ prediction for $\mathscr{S}_5$ based on $\mathscr{S}_3$ & $\mathscr{S}_4$.

Figure 5: Multivariate $\mathscr{TH}$ predictions.

model; instead, we compared the time needed to generate output once the TCN model was trained against the PDEVS MiniFab model. The results are as summarized in Table 3. The TCN model consistently takes a fixed amount of time to execute once trained (time complexity $= O(1)$), while the PDEVS simulation requires executions of the diffusion, implantation, and lithography models, resulting in higher execution times due to larger lot sizes.

Table 3: TCN vs. PDEVS performance evaluation.

	Execution Time (seconds)		**Execution Time Ratio**
LOT SIZE	**PDEVS**	**TCN**	**PDEVS/TCN**
Small	96	2	48
Medium	354	2	177
Large	868	2	434

6 CONCLUSION & FUTURE WORK

The research in this paper suggests that ML/DL models can be used to generate temporal dynamics of semiconductor manufacturing systems that are close to those of PDEVS simulation models. Deep Learning models presented in this paper show that the predicted throughput and turnaround time series could have acceptable accuracy given that the factory dynamics have prescribed operational scenarios. In many cases, we observe a reasonable accuracy with R^2 scores to be within 0.8 to 0.99. The mean square error values vary for throughput and turnaround time, which can be attributed to the actual values being small ($\approx e^{-8}$) and large ($\approx 10^3$), respectively, but can be used as a comparative measure to analyze the predictive capabilities of statistical and DL models. TCN predictions show higher throughput and turnaround time accuracies compared with those of ARIMA, RNN, and LSTM and have better mean absolute percentage error scores for the 8-stage cascade MiniFab. Also, the multivariate analysis suggests time series models can account for the effects of throughput and turnaround time of preceding stages of the cascade MiniFab to predict the values of a current stage. Future work includes the continuation of the applicability of deep learning models for large-scale, complex semiconductor factories, particularly for fast what-if purposes. Other research directions include using machine learning models to assist in developing PDEVS models and digital twins in operational manufacturing systems.

ACKNOWLEDGMENTS

This research is funded by Intel Corporation, Chandler, Arizona, USA. We are grateful to the reviewers who provided constructive critiques and suggestions on an earlier version of this paper.

REFERENCES

ACIMS. 2023. “DEVS-Suite Simulator, version 7.0.0”. https://acims.asu.edu/devs-suite/, accessed: 10th January 2024.

Atalan, A., H. Şahin, and Y. A. Atalan. 2022. “Integration of Machine Learning Algorithms and Discrete-Event Simulation for the Cost of Healthcare Resources”. *Healthcare* 10(10):1920.

Bai, S., J. Z. Kolter, and V. Koltun. 2018. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. July 10th-15st, Stockholmsmässan, Sweden, 732-740.

Bartos, B. J., M. Mioduszewski, M. Renner, and R. McCleary. 2017. “An Application of Discrete Event Simulation for Planning and Resource Allocation in a State Hospital System Servicing Both Criminal and Civil Commitments”. In *2017 Winter Simulation Conference (WSC)*, 4509–4511 https://doi.org/10.1109/WSC.2017.8248182.

Box, G. E. P. and G. M. Jenkins. 1970. *Time Series Analysis:Forecasting and Control*. 4th ed. Hoboken: John Wiley & Sons.

Chan, K., M. Rabaev, and H. Pratama. 2022. “Generation of Synthetic Manufacturing Datasets for Machine Learning Using Discrete-Event Simulation”. *Production & Manufacturing Research* 10(1):337–353.

Chien, C.-F., H. Ehm, J. W. Fowler, K. G. Kempf, L. Mönch, and C.-H. Wu. 2023. “Production-Level Artificial Intelligence Applications in Semiconductor Supply Chains”. *IEEE Transactions on Semiconductor Manufacturing* 36(4):560–569.

Choi, S. J. and T. G. Kim. 2002. “Identification of Discrete Event Systems Using the Compound Recurrent Neural Network: Extracting Devs From Trained Network”. *Simulation* 78(2):90–104.

Chow, A. C. H. and B. P. Zeigler. 1994. “Parallel Devs: A Parallel, Hierarchical, Modular Modeling Formalism”. In *1994 Winter Simulation Conference (WSC)*, 716–722 https://doi.org/10.1109/WSC.1994.717419.

Cimino, C., E. Negri, and L. Fumagalli. 2019. “Review of Digital Twin Applications in Manufacturing”. *Computers in Industry* 113:103130.

Ehm, H. and T. Ponsignon. 2012. “Future Research Directions for Mastering End-To-End Semiconductor Supply Chains”. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. August 20th-24th, Seoul, South Korea, 641-645.

Fronckowiak, D., A. Peikert, and K. Nishinohara. 1996. “Using Discrete Event Simulation to Analyze the Impact of Job Priorities on Cycle Time in Semiconductor Manufacturing”. In *IEEE/SEMI 1996 Advanced Semiconductor Manufacturing Conference and Workshop*. November 12th-14th, Cambridge, MA, USA, 151-155.

Goldman, C. V., M. Baltaxe, D. Chakraborty, and J. Arinez. 2021. “Explaining Learning Models in Manufacturing Processes”. *Procedia Computer Science* 180:259–268.

Griffiths, F. and M. Ooi. 2018. “The Fourth Industrial Revolution-Industry 4.0 and Iot [Trends in Future I&M]”. *IEEE Instrumentation & Measurement Magazine* 21(6):29–43.

Hastie, T., R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. New York: Springer.

Hintze, J. L. and R. D. Nelson. 1998. “Violin Plots: A Box Plot-Density Trace Synergism”. *The American Statistician* 52(2):181–184.

Hochreiter, S. and J. Schmidhuber. 1997. “Long Short-Term Memory”. *Neural Computation* 9(8):1735–1780.

Ingemansson, A. and J. Oscarsson. 2005. “Discrete Event Simulation and Automatic Data Collection Improve Performance in a Manufacturing System”. In *2005 Winter Simulation Conference (WSC)*, 1441–1445 https://doi.org/10.1109/WSC.2005.1574410.

James, B. and B. Yoshua. 2012. “Random Search for Hyper-Parameter Optimization”. *Journal of Machine Learning Research* 13(10):281–305.

Kampa, A., G. Gołda, and I. Paprocka. 2017. “Discrete Event Simulation Method as a Tool for Improvement of Manufacturing Systems”. *Computers* 6(1):10.

Lang, S., F. Behrendt, N. Lanzerath, T. Reggelin, and M. Müller. 2020. “Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production”. In *2020 Winter Simulation Conference (WSC)*, 3057–3068 https://doi.org/10.1109/WSC48552.2020.9383997.

Lingitz, L., V. Gallina, F. Ansari, D. Gyulai, A. Pfeiffer, W. Sihn *et al*. 2018. “Lead Time Prediction Using Machine Learning Algorithms: A Case Study by a Semiconductor Manufacturer”. *Procedia Cirp* 72(5):1051–1056.

Mazzei, D. and R. Ramjattan. 2022. “Machine Learning for Industry 4.0: A Systematic Review Using Deep Learning-Based Topic Modelling”. *Sensors* 22(22):8641.

Mehdiyev, N., D. Enke, P. Fettke, and P. Loos. 2016. “Evaluating Forecasting Methods by Considering Different Accuracy Measures”. *Procedia Computer Science* 95(4):264–271.

Qiao, D. and Y. Wang. 2021. “A Review of the Application of Discrete Event Simulation in Manufacturing”. *Journal of Physics: Conference Series* 1802(2):022066.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. “Learning Representations by Back-Propagating Errors”. *Nature* 323(6088):533–536.

Saadawi, H., G. Wainer, and G. Pliego. 2016. “Devs Execution Acceleration With Machine Learning”. In *2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS)*. April 3rd-6th, Pasadena, CA, USA, 1-6.

Sarjoughian, H. S., F. Fallah, S. Saeidi, and E. J. Yellig. 2023. “Transforming Discrete Event Models to Machine Learning Models”. In *2023 Winter Simulation Conference (WSC)*, 2662–2673 https://doi.org/10.1109/WSC60868.2023.10407348.

Sarker, I. H. 2021. “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions”. *SN Computer Science* 2(6):420.

Seok, M. G., C. W. Chan, W. Cai, H. S. Sarjoughian, and D. Park. 2020. “Runtime Abstraction-Level Conversion of Discrete-Event Wafer-Fabrication Models for Simulation Acceleration”. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. June 15th-16th, Miami, Florida, USA, 83-92.

Singgih, I. K. 2021. “Production Flow Analysis in a Semiconductor Fab Using Machine Learning Techniques”. *Processes* 9(3):1–18.

Spier, J. and K. Kempf. 1995. “Simulation of Emergent Behavior in Manufacturing Systems”. In *Proceedings of SEMI Advanced Semiconductor Manufacturing Conference and Workshop*. November 13th-15th, Cambridge, MA, USA, 90-94.

Subramaniyan, M., A. Skoogh, H. Salomonsson, P. Bangalore, and J. Bokrantz. 2018. “A Data-Driven Algorithm to Predict Throughput Bottlenecks in a Production System Based on Active Periods of the Machines”. *Computers & Industrial Engineering* 125:533–544.

Tanious, R. and R. Manolov. 2022, 09. “Violin Plots as Visual Tools in the Meta-Analysis of Single-Case Experimental Designs”. *Methodology European Journal of Research Methods for the Behavioral and Social Sciences* 18(3):221–238.

von Rueden, L., S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese *et al*. 2021. “Informed Machine Learning–a Taxonomy and Survey of Integrating Prior Knowledge Into Learning Systems”. *IEEE Transactions on Knowledge and Data Engineering* 35(1):614–633.

Xie, C. and T. T. Allen. 2015. “Simulation and Experimental Design Methods for Job Shop Scheduling With Material Handling: A Survey”. *The International Journal of Advanced Manufacturing Technology* 80(1):233–243.

AUTHOR BIOGRAPHIES

VAMSI KRISHNA PENDYALA is a Ph.D. student in the Computer Science program in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University (ASU), Tempe, AZ, USA. He can be reached at vpendya2@asu.edu.

HESSAM S. SARJOUGHIAN is an Associate Professor of Computer Science and Computer Engineering in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University (ASU), Tempe, Arizona. His research interests include model theory, poly-formalism modeling, collaborative modeling, simulation for complexity science, and M&S frameworks/tools. He is the co-director of the Arizona Center for Integrative Modeling and Simulation https://acims.asu.edu. He can be contacted at hessam.sarjoughian@asu.edu.

EDWARD J. YELLIG is the director of Operational Decisions Support Technology at Intel Corporation. He has been with Intel for 26 years and has a Ph.D. in Operations Research with an emphasis on discrete event modeling of large-scale systems. His focus has been on developing fab models for determining capital requirements and is also responsible for the real-time digital twin tactical models. He can be contacted at edward.j.yellig@intel.com.