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.

VALIDATION TOWARDS REALISTIC SYNTHETIC DATASETS IN PRODUCTION PLANNING

Jan Michael Spoor[1], Marvin Matthes[2], Martin Krockert[2], and Jens Weber[3]

[1]Institute for Information Management in Engineering, Karlsruhe Institute of Technology, GERMANY
[2]Dept. of Informatics & Mathematics, University of Applied Sciences Dresden, GERMANY
[3]Faculty of Technology, Baden-Wuerttemberg Cooperative State University, Lörrach, GERMANY

ABSTRACT

For large-scale simulations, a sufficient data amount is required. Despite an increasing data availability, it is still challenging to gather large-scale datasets, which are comprehensive, correct, accessible, and realistic, to validate new algorithms and models. An alternative is the use of synthetic data. Thus, we propose a novel methodology to generate realistic datasets. Based upon the statistical properties of real-world data, synthetic datasets are generated by ML models and filtered for anomalous values. The generated datasets are then compared to find the most suitable one. For this validation procedure, a modified Hopfield neural network model is extended to enable an analysis of sequences and to derive a comparison metric. The method demonstrates its applicability by providing an in-depth comparison of all tested data generators using a real-world dataset of a mid-size manufacturing company, whereby transformer-based generators proved most suitable. More diverse use cases should be evaluated in future research.

1 INTRODUCTION

The ability to run simulations is an important part of the planning, commissioning, operation, and modernization of manufacturing systems. Simulations proved themselves to be cost-efficient, are fast in execution, provide insightful results, and do not yield any additional risks (Mourtzis 2020). As input, simulations require a large body of real-world data to provide meaningful results, in particular when using data-driven methods from the field of Artificial Intelligence (AI) and Machine Learning (ML).

In theory, more data is currently available today than ever before. However, distributed data ownership and prohibitive data collection often reduces the ability to use large amount of data for simulations, i.e., Arinez et al. (2020) and Wuennenberg et al. (2023) state that it can be very difficult for researchers to obtain large amounts of data due to an operational prohibitive wide-scale data collection and restrictions of data access and exchange due to security concerns. Thus, practitioners must rely on their data collection, which is often costly and time-intense. Therefore, the ability to gather more meaningful data to enhance or enlarge available datasets without real-world operations becomes important. As a possible solution, databases from dedicated libraries for training, testing, or to generate synthetic data can be used. While data libraries offer large datasets, which are validated for correctness and often sufficiently comprehensive, they are limited in their specific use case and the specific distribution of feature values. The generation of synthetic data circumvents the costly or sometimes impossible real-world data collection and can be tailored for all use cases' specific requirements regarding distribution of values and features. In addition, synthetic data are comprehensive and can be generated in any scope or amount Völker et al. (2001).

In contrast to the advantages of the generation procedure, the validation of synthetic datasets for realism is difficult since the data amount generated tends to be large and only aggregated Key Performance Indicators (KPI), often describing the distribution, e.g., average and frequency, of one or multiple features, are evaluated. However, KPIs for individual distribution might be misleading and do not guarantee plausibility and a detailed evaluation without aggregated KPIs might be too time-consuming and require manual input. Therefore, the research question of this paper is how a fast and comprehensive data validation which does

not solely rely on aggregated KPIs can be integrated into a data generation approach to test the plausibility of synthetic data using the limited amount of available real-world data as a benchmark. In addition, a correction procedure to detect non-plausible data records is required to improve a given synthetic dataset. The proposed methodology builds on the data generation approach by Matthes et al. (2023) and adds the embedding of modified Hopfield neural networks introduced by Spoor and Weber (2024) to filter non-plausible data and to validate different data generation approaches.

This paper starts by discussing notable methods and approaches of data generation and validation from the body of literature in Section 2. Subsequently, the proposed methodology is presented in Section 3 and the results using the proposed approach in a real-world use case are discussed in Section 4. Lastly, the methodology is discussed in Section 5 and a conclusion is provided in Section 6.

2 LITERATURE REVIEW

Generating data that accurately simulates the complexity of real-world production systems is a significant challenge in the realm of production planning and control. To effectively mimic the intricate interdependencies and patterns inherent in actual production environments, innovative approaches are necessary. As common use cases of synthetic data, Libes et al. (2017) name the training of ML models, esp. the verification of computer models based on their mathematical equations and solutions, the validation of how accurately they represent the underlying real-world application, the optimization of the real-world entity based on the results achieved by analysis with synthetic data, and the augmentation of real-world data, e.g., by replacing missing data during a data collection.

Different authors demonstrate the generation of synthetic data using ML methods or statistical approaches leveraging domain knowledge, e.g., Mannino and Abouzied (2019). Approaches such as those presented by Fernandes et al. (2020) and Adolphy et al. (2015) use synthetic data generation which involves creating realistic, synthetic copies of production data. This method leverages algorithms to replicate the complex structures and correlations, found in real data. The benefits of using synthetic data are numerous. These include compliance with privacy regulations, the ability to share data across protected institutions, and the potential to speed up development cycles. Synthetic data is particularly valuable because it accurately mimics the properties of real machine sequences, making it an invaluable tool for validating production planning algorithms through sophisticated simulations.

Generating synthetic data for production systems is challenging due to the unique nature of production master data. Despite its advantages, generating synthetic data for production systems is not straightforward. Production master data includes a wide range of attributes and relationships specific to the manufacturing process, such as machine sequences, production schedules, and material requirements. Generating realistic synthetic data can be challenging due to the complexity of production systems. Basic tools and approaches like data anonymization libraries often fail to capture the nuanced interdependencies of these systems.

To the authors' best knowledge, very little research is conducted in the development of validation procedures testing if synthetic datasets are suitable representations of the underlying real-world application. For example Krockert et al. (2021) note that synthetic data often only represent the real-world data regarding KPIs defining their structure, but they may lack realism. Nevertheless, Krockert et al. (2021) show that machinery sequences in manufacturing can sufficiently be simulated using synthetic data. Matthes et al. (2023) suggest using statistical KPIs, such as the average length of a machine sequence calculated using the arithmetic mean, to evaluate the authenticity of machine sequences in production. They also recommend calculating the relative frequency of occurrence for subsequences within machine sequences, similar to probability distribution functions, to assess the distribution and frequency of specific patterns within the data. Other statistical approaches include Jaccard similarity (Jaccard 1901), which assesses overlap, Levenshtein distance, which measures similarity between sequences, and K-gram indexing, which identifies common subsequences. These methods, traditionally used in fields such as text processing, can be customized to analyze production sequences and offer an understanding of similarities and differences between generated synthetic data and real data.

3 METHODOLOGY

As foundation of the proposed methodology, Matthes et al. (2023) demonstrate the generation of synthetic data using a Bayesian network, a transformer network from the field of deep-learning architectures, and a statistical approach for the generation process. However, the validation and comparison of the synthetic dataset is only conducted based on aggregated KPIs. Thus, the methodology cannot validate if all machinery sequences in the synthetic dataset are realistic or if certain kinds of sequences are accurately represented regarding quality and quantity. Modified Hopfield neural networks proposed by Spoor and Weber (2024) are hereby utilized in two ways: firstly, they enable anomaly detection of the generated data, and secondly, they can validate and compare the generated data based on modeled directed multigraphs with loops representing the sequential information. Thus, this method enables a more in-depth analysis of the data compared with aggregated structural KPIs. In addition, the real-world data, which is also used for the data generation, is applied for training the modified Hopfield neural network. Modified Hopfield neural networks are also useful in this case, since the generation of synthetic data is based only on correct data.

The novel methodology from this research, illustrated in Figure 1, incorporates a data filtering step. This step uses a modified Hopfield neural network to eliminate outliers and standardize data, enhancing the quality of synthetic machinery sequences. A second novelty of this research is the validation procedure using a metric derived from a modified Hopfield neural network and the extension of the model to enable the analysis of sequences in addition to combinations.

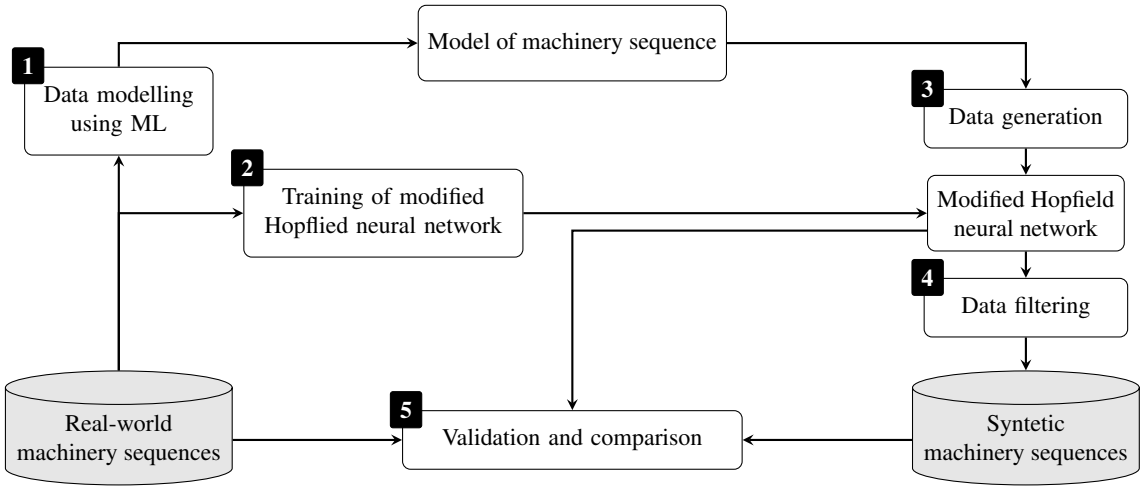


Figure 1: Schematic methodology to generate, validate, and compare synthetic data utilizing a modified Hopfield neural network for validation and filtering of anomalous synthetic data.

In the first step, the underlying real-world data set is utilized by data modeling using ML methods to set up a model of the machinery sequences (1). In addition, the real-world data is used to train the modified Hopfield neural network, which only requires true negative instances, i.e., only correct real-world data without incorrect examples (2). The model of machinery sequences is then used to generate synthetic datasets (3). Thereafter, the generated synthetic datasets are filtered by the anomaly detection capabilities of the modified Hopfield neural network. Thus, the final synthetic dataset per each method of data generation is corrected by excluding unlikely, uncommon, or anomalous data points (4). To subsequently validate the datasets, the generated synthetic datasets are compared to the real-world data to decide which dataset comes closest to reality and represents the real-world. This is again conducted by the classification, resp. clustering capabilities of the modified Hopfield neural network (5). As a result, the methodology will recommend one certain dataset from a set of multiple datasets using different generation models or generation model parameters. The recommended dataset will describe the real-world data most suitable.

3.1 Data Generation

3.1.1 Statistical-based Approaches

Machine transition matrices or graphs are standard models for the abstract description of sequences of machines (Schuh 2007). In this paper, they are utilized as a reference for the application of Bayesian networks and transformers. A machine transition matrix is a square matrix, with its elements $\pi_{i,j}$ indicating the probability of transition from machine i to machine j.

As illustrated in Figure 2a, a simple machine transition matrix is represented for three machines. New machine sequences are generated by sampling over the transition matrix with a given randomized sequence length, whose distribution can be extracted from the real data. This approach is abbreviated as ST in the further discourse of this paper. Figure 2b shows an extended approach (abbreviated as TASS) where a source and sink are added to the matrix. Therefore, the sampling will always start in the source row (Q) and stop sampling at the sink row (S). Hence, the sequence length will be determined by the matrix itself (Krockert et al. 2021).

	M1	M2	M3
M1	0,16	0,47	0,37
M2	0,31	0,25	0,44
M3	0,56	0,29	0,15

(a) [ST] Simple transition matrix

	M1	M2	M3	S
Q	0,49	0,35	0,16	0,00
M1	0,16	0,42	0,30	0,12
M2	0,08	0,17	0,44	0,31
M3	0,17	0,11	0,20	0,52

(b) [TASS] Transition matrix with source (Q) and sink (S)

Figure 2: Statistical sequence generation with a machine transition matrix.

As Matthes et al. (2023) describes, a significant limitation of this model lies in its inability to adequately represent recurring sub-sequences of machines, as the probability of transition from machine i to machine j depends solely on machine i. Consequently, potential patterns in preceding and subsequent sequences of machines remain unaccounted for.

3.1.2 Transformer-based Approaches

Matthes et al. (2023) applies a Transformer-based approach with an autoregressive architecture to generate machine sequences. The Transformer architecture was proposed by Vaswani et al. (2017) and has established itself for a variety of applications in natural language processing and beyond. The autoregressive approach, introduced and further developed by the works of Radford et al. (2019) and Brown et al. (2020), is a neural network that processes sequential data by learning a probability distribution over possible next elements in the sequence, using previous elements as context, highlighted in Figure 3.

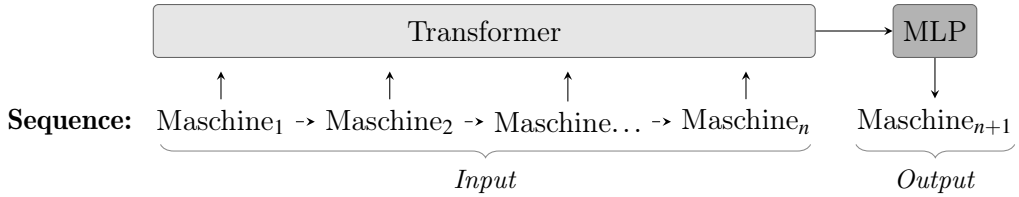


Figure 3: Simplified transformer architecture.

Transformers utilize self-attention to capture and weight relationships between different positions within the input sequence. Thus, they can predict the next element in the sequence based on the preceding elements.

$$P(x_t|x_1,x_2,\ldots,x_{t-1}) = \text{Transformer}(x_1,x_2,\ldots,x_{t-1};\theta)$$

Here, x_t represents the $t-th$ element of the sequence, and $[x_1,x_2,\ldots,x_{t-1}]$ are the preceding elements in the sequence. The autoregressive Transformer uses the preceding elements as context for prediction. To generate the probability distribution for the next element x_t, the output vector of the Transformer is classified by a Multilayer Perceptron (MLP). Thereby, θ denotes the set of weights of the Transformer. During the training process, the weights are adjusted to optimize the Transformer's performance in predicting from the data. The adjusted weights enable the Transformer to recognize patterns and relationships in the input sequences and make corresponding predictions for the next elements of the sequence.

3.1.3 Bayesian Network-based Approaches

A Bayesian network is an acyclic graph in which nodes represent random variables and edges represent direct (causal) dependencies between the random variables. Each node or random variable is associated with probability distributions for every combination of states of the parent nodes, indicating the likelihood of the states of the random variables given the states of the parent nodes. Bayesian networks can be trained and used for tasks such as forecasting, diagnosis, and sensitivity analysis under uncertainty. Additionally, it is possible to draw samples from Bayesian networks that statistically correspond to the data from which the Bayesian networks were created.

Matthes et al. (2023) use a Bayesian network to learn machine sequences from real production, and subsequently, samples of new machine sequences are drawn from the Bayesian networks. The Bayesian network is modeled as follows: each node of the network represents positions of work operations, and each work operation position can assume states representing the machines occurring at that position. This Bayesian network representation allows the modeling of the machine sequence. Since Bayesian networks encode multiple conditional probabilities, it will generate frequently occurring sub-machine sequences.

Learning is performed using the score-based hill-climbing approach (Tyugu and Tyugu 2007) and the Bayesian information criterion (BIC) score as a quality measure (Koller and Friedman 2009). For the subsequent generation of machine sequences from the Bayesian network, forward sampling is employed (Koller and Friedman 2009).

3.2 Data Validation

Modified Hopfield neural networks proposed by Spoor and Weber (2024) are originally designed to analyze combinations. Thus, the network layout is adjusted to analyze sequences. Base for the adjusted network layout is the visualization of the sequential data as a directed multigraph with loops.

In modified Hopfield Neural Networks, sequences of objects from an amount of V different object types are modeled using a $V \times V$ connection matrix M. Each element m_{ij} of the connection matrix indicates either an active edge of the graph or no active edge between an object of type i and of type j.

$$m_{ij} = \begin{cases} +1, & \textit{active edge} \\ -1, & \textit{inactive edge} \end{cases}$$

At first, data used for training and testing is transformed into connection matrices. For each sequence of object types $\mathscr{C}$ a connection matrix $\mathbf{M}$ is computed. An entry $m_{ij} = -1$ indicates no edge between vertex i and vertex j of the graph. If combinations are analyzed, the connection matrix will be symmetrical. For sequences, a connection $m_{ij} = 1$ indicates that object type j is a successor of type i in the given sequence and the connection matrix is no longer symmetrical.

The relation between the object types representing the sequence can be expressed by a directed multigraph with loops, where the vertices are the object types and the weight of the edges represents the likelihood of a sequence between object type i and object type j. The graph of the likelihood of a sequencing is represented by a $V \times V$ weight matrix W using the weights w_{ij} for each sequence between vertex i and j.

$$-1 < w_{ij} < +1 \ \forall i, j$$

A negative weight represents a negative correlation between these vertices, i.e., these two object types are most likely not successive. A positive weight represents a common sequence of vertices.

The network of weights is stimulated by an input of a connection matrix, and the weights are iteratively updated to decrease the networks energy while stimulated. The energy of the stimulated network is calculated using an adaptation from the energy computation in Hopfield networks (Hopfield 1984) or resp. in the Ising model (Brush 1967) applying the active or inactive connection m_{ij} instead of the individual active or inactive vertices. This enables loops by circumventing the restriction that $w_{ii} = 0$. The energy resulting from the weight network when stimulated by a connection matrix using a correction value θ and the Kronecker delta function $\delta[n]$ to count the active vertices for sequences is given as follows:

$$H = -\sum_{i=0}^{V}\sum_{j=0}^{V} w_{ij}m_{ij} - \theta * \delta[m_{ij} - 1]$$

An equilibrium energy of the network without stimulation by a connection matrix is defined as follows:

$$H_{eq} = -\sum_{i=0}^{V}\sum_{j=0}^{V} w_{ij}^2$$

This equilibrium is equivalent to the Ising model without external forces applied. The weight w_{ij} is hereby used as an estimator of the mean probability if a connection is active. For combinations, the summarization is applied over the triangular matrix and diagonal values.

The correction value is the average effect on the system energy if any combination is active. A negative correction value indicates that only a limited number of combinations are active at the same time, while a positive correction value indicates that combinations are more likely to be active than inactive.

$$\theta = \frac{1}{V^2}\sum_{i=0}^{V}\sum_{j=0}^{V} w_{ij}$$

For each element of the training set, the network gets stimulated and is optimized to reduce the energy of the stimulated state in the direction of the equilibrium energy (minus the correction value). The energy difference between the equilibrium and the stimulated state is given as follows:

$$H - H_{eq} = -\sum_{i=0}^{V}\sum_{j=0}^{V} w_{ij}(m_{ij} - w_{ij}) - \theta * \delta[m_{ij} - 1]$$

To optimize the difference in energy between the two states by updating the weight, a gradient is set up to update each weight individually using a training rate $\alpha \in (0,1)$ as step sizes of adaption per iteration t. The weight update can be adjusted using a decay and reduction of the training rate for regularization.

$$w_{ij}^t = w_{ij}^{t-1} + \alpha\left(m_{ij} - w_{ij}^{t-1}\right)$$

Firstly, this network is utilized for anomaly detection by comparing the stimulated energy $H(\mathscr{C})$ of a sequence $\mathscr{C}$. The rational is that common sequences have lower stimulated energies than uncommon sequences. If multiple sequences are analyzed, the sequences with lower energies are more common occurrences of sequences, while sequences with higher stimulated energies are either uncommon or incorrect. The specific energy level depends on the weight matrix and thus, the energies must be evaluated in comparison to a real-world testing set of sequences. Using the average stimulated energy of the

training data and their standard deviation σ, a critical energy H_{crit} is defined. Sequences with stimulated energies above this critical energy are considered anomalous. The critical energy can be selected so that $H_{crit} > \overline{H} + c * \sigma$ given a confidence interval c. Only a one-sided evaluation of this confidence interval is necessary, since low-energy states are assumed to be correct.

Secondly, it is possible to use the network layout to compare different classes of sequences, in this case different types of data generation models as part of the validation and comparison of the methodology in Figure 1. Hereby, the trained weight matrices per class can be used. Each class k yields a specific weight matrix $\mathbf{W}_k$ which is computed by training only with sequence data from this class. The difference between two classes k and l is measurable by comparing the individual entries of the weight matrices and squaring the results. Thus, the delta weight matrix $\Delta\mathbf{W}_{kl}$ between class k and l using the Hadamard product is computed as follows:

$$\Delta\mathbf{W}_{kl} = (\mathbf{W}_k - \mathbf{W}_l) \odot (\mathbf{W}_k - \mathbf{W}_l)$$

An element of the delta weight matrix w_{ijkl} close to zero indicates that the combinations between object i and j are as common (or uncommon) in classes k and l. Values larger than zero, in particular values larger 1, will indicate combinations which are very likely in one class but highly unlikely in the other. Therefore, these are the sequences which separate the classes. If two classes yield entries in their weight matrices which are very close, they will be more difficult or even impossible to separate during anomaly detection. On the other hand, classes with high delta weight values w_{ijkl} for all combinations will be easily separable by the anomaly detection method. Thus, classes with overall high delta weight values are more different to each other and classes with low high delta weight values are more similar to each other.

To compute a dissimilarity, the Frobenius norm $\| \bullet \|_F$ is applied. For a number of K classes, a $K \times K$ distance matrix $\mathbf{D}$ for the dissimilarity between all classes is computed. The distance matrix hereby yields at position k and l the corresponding entries $d(k,l) = \|\mathbf{W}_k - \mathbf{W}_l\|_F$. Using this distance matrix, the classes can be formally compared and evaluated. For combinations (with a symmetric weight matrix), an adjusted Frobenius norm is applied, which summarizes only the upper triangular matrix and diagonal values.

In addition, the stimulated energy levels can be compared using the network with the weight matrix trained by real-world data as a baseline. This requires the comparison using the Wasserstein Metric W between the resulting energy distributions. The first-order Wasserstein metric W_1 can be visualized as the Earth-Movers-Distance (EMD) where one bar chart is viewed as piles of earth which must be transported to holes of the other bar chart. The distance is the least amount of work required to fill the holes, while the work is defined by the amount of sand times the transported ground distance of the sand (Rubner et al. 2000). The EMD is the solution which minimizes this transport problem. The ground distance between bins of the distribution is hereby evaluated using the L_1 metric. Therefore, the distribution must priorly be discretized. The Wasserstein metric is evaluated for each pair of energy distributions from a total amount of K analyzed datasets. Thus, this results also in a symmetric distance matrix $\mathbf{D}$ which is used to evaluate the dissimilarity of the energy distribution of two datasets.

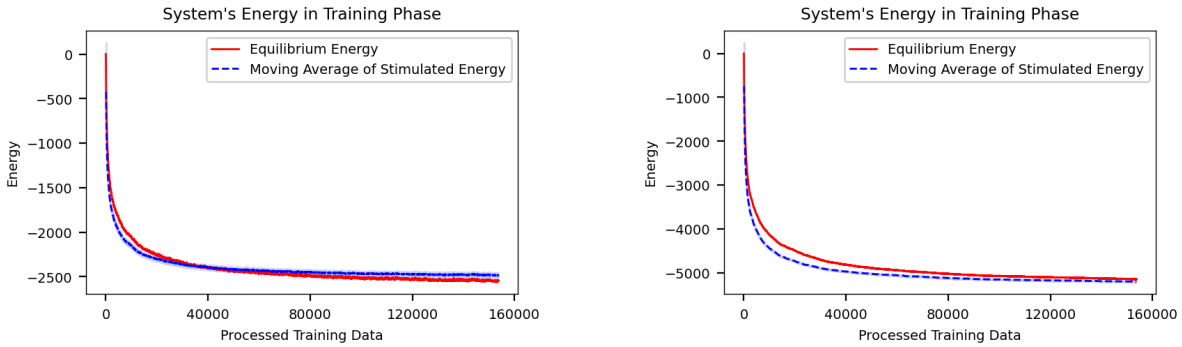
Both methods for computing the distance have different advantages and disadvantages. The Frobenius norm between the weight matrices directly measures the sequence likelihood over the whole dataset. Thus, changes in the likelihood of sequences are directly measured by this evaluation. However, this requires the training of all classes to converge and thus, each dataset must have sufficient data points available. On the other hand, the Wasserstein metric between the energy distributions measures the similarity of the complexity of the data. A synthetic dataset might have an overrepresentation of common cases which individually fit exactly with the weights but result in energy distribution differences. In this case, the distance from the Frobenius norm between the weight matrices would be small, but the distributions would look quite different and might result in a large distance using the Wasserstein metric. *Vice versa* the real-world data could yield some rare but correct cases which are represented in the resulting weight matrix but are so seldom that they are not reflected as modes of the energy distribution. In this case, the distance between the energy distributions is very small, but the distance between the weight matrices is larger.

4 VALIDATION OF METHODOLOGY

To validate the proposed approach a dataset from a manufacturing company is used containing 170787 real-world workstation sequences. The data was collected between 2022 and 2023 and covers all processes of one production facility. In total, 73 different workstations are covered within the dataset. The number of workstations in one sequence is between 1 (a single workstation is used in the process) up to 77, whereby workstations can occur multiple times in the same record. This real-world dataset is used to generate four synthetic datasets using SA and TASS, a Bayesian network, and a transformer-based approach.

For the validation and comparison of the synthetic datasets, the methods are evaluated by their (a) capability of generating correct workstation combinations, i.e. the right workstations are paired in a record, but the sequence is neglected, and (b) capability of generating correct sequences. The first evaluation tests for incorrect workstation combinations. The second evaluation tests for correct successor and predecessor relations of single workstations, but neglects the relation of not immediately succeeding workstations. Thus, both evaluations should be carefully considered.

To conduct the validation procedure, 90% of the real-world data is used to train the modified Hopfield neural network. The remaining 10% are used to compute the stimulated energies as a comparison base to the synthetic datasets, similar to a hold-out validation. Firstly, each sequence is converted into a 73×73 connection matrix. For the training procedure, the training rate is selected as $\alpha = 0.05$ and no regularization terms are applied. Two networks, (a) using combinations and (b) using sequences, are separately trained. The result of the training procedure is given in Figure 4.



(a) Combinations

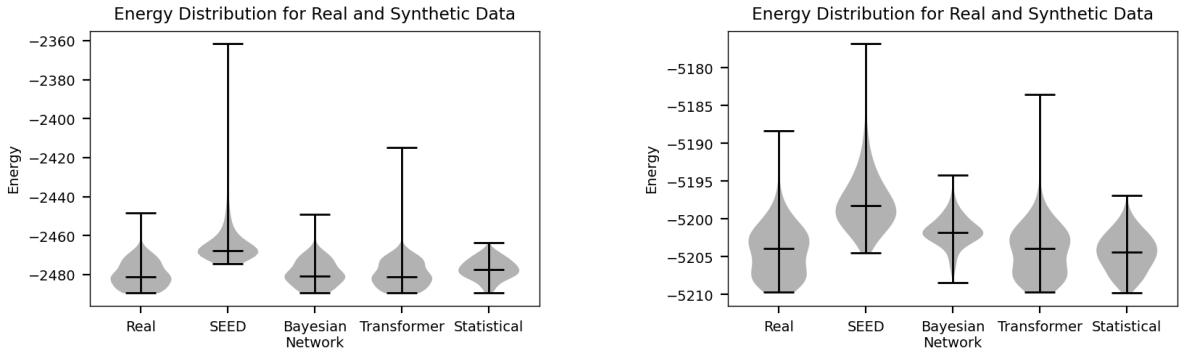
(b) Sequences

Figure 4: Training procedure using the moving average of the stimulated energy over 100 iterations, its 3-standard derivation area, and the equilibrium energy.

The training converges in both cases at around 40000 records to a stable equilibrium and thus, the amount of training data is sufficient for setting-up the modified Hopfield neural network.

To conduct a test for anomalous generated data, the energy distributions by stimulating the trained network with the synthetic data (separately for combinations and sequences) are compared in Figure 5.

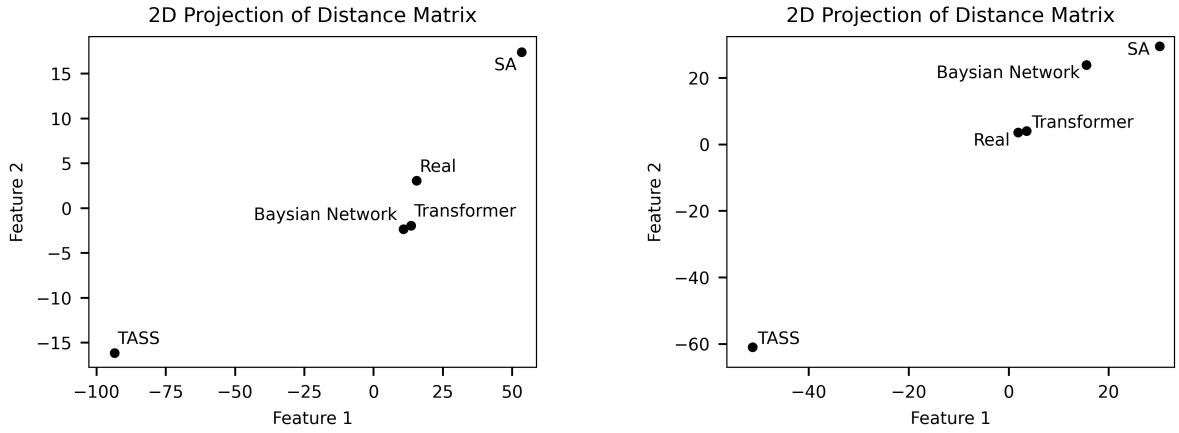
As visible in the violin plots of the distributions, only a small amount of the synthetic data records yields an energy above the maximum energy of the real-world data. When using the maximum energy of the real-world dataset as a threshold H_{crit} for anomaly detection in combinations and sequences, neither the Bayesian network nor the SA approach detects any anomalies. The transformer-based approach yields a ratio of 0.004% anomalies for the evaluation of combinations and 0.005% for the evaluation of sequences. Thus, there exist rare cases where the transformer-based approach generates anomalous data records. The approach using TASS generates 5.13% anomalous data records for combinations and resp. 1.54% for sequences. Therefore, this approach would require the most stringent filtering of anomalous data.



(a) Combinations (b) Sequences

Figure 5: Distribution of the real-world and synthetic datasets' energy levels displayed as violin plots.

For the validation of the generated data, the distributions in Figure 5 can be visually compared, in particular with the distribution of the real-world data. The tendency by TASS and in rare cases by the transformer-based approach to generate anomalous data is directly visible by comparing the upper maximum limits of the box plots. In addition, it is visible that TASS is more likely to generate data in the higher quantile of the real-world data energies and all energies of generated combinations (and most in the case of sequences) using TASS exceed the median of the real-world data energy level. This comparison can be evaluated using the Wasserstein metric. Each distribution is hereby discretized using 200 bins and the Wasserstein metric is computed by the python library POT provided by Flamary et al. (2021). To visualize the results a 2D projection of the distance matrix is computed using the sklearn library by Pedregosa et al. (2011). The resulting projection of the distance matrix is shown in Figure 6.



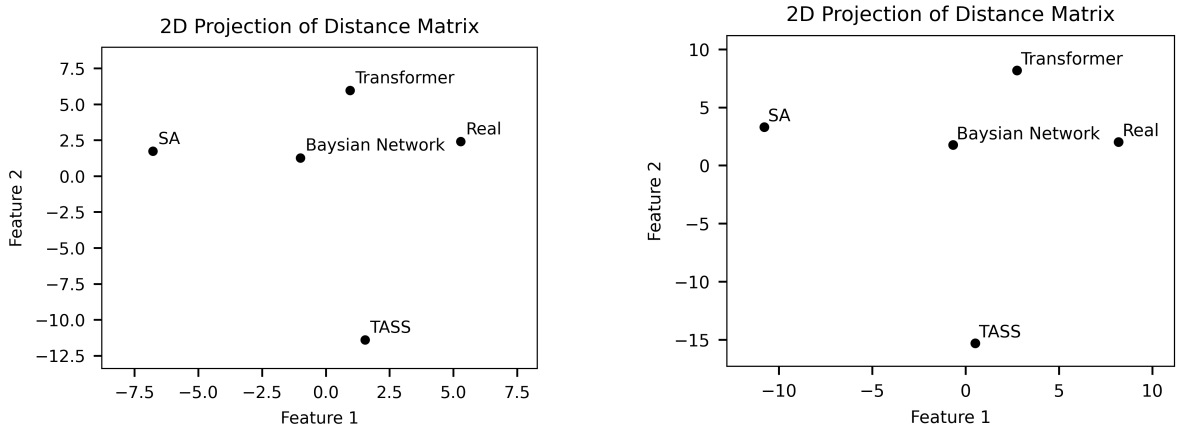
(a) Combinations (b) Sequences

Figure 6: 2D projection of the resulting distance matrix between the energy distributions in Figure 5 of the real-world dataset and the synthetic datasets using the Wasserstein metric.

As visible, the energy distributions of the transformer-based and Bayesian network-based data generation are close to the real-world data. In the case of sequence analysis, the distribution of the transformer-based methods is nearly identical with the real-world data energy distribution. The similarity of the distributions is also visible in Figure 5, besides the small number of outliers. Thus, the comparison of the energy

distributions indicates that the transformer-based methods are most suitable for a data generation if outliers are filtered. Bayesian networks are also viable alternatives, in particular if only combinations are observed.

In addition to the analysis using the Wasserstein metric, the data generation approaches are compared using the Frobenius norm between individually trained weight matrices. The results are given in Figure 7.



(a) Combinations

(b) Sequences

Figure 7: 2D projection of the resulting distance matrix between the weight matrices of the real-world dataset and the synthetic datasets using the Frobenius norm.

Using the Frobenius norm, the transformer-based approaches and the Bayesian networks perform well in both analyses. Most notably, the overall results for both metrics and for both evaluations of either combinations or sequences are very similar. This indicates that the analysis is quite robust and both applied metrics are comparable. The results align with the KPI-based evaluation by Matthes et al. (2023), but the applied novel method implemented in this research adds an additional detail to the validation procedure and thus enhance the generated data for the real-world scenarios described in Matthes et al. (2023).

5 DISCUSSION

The proposed approach yields some advantages in a practical application. Firstly, the required amount of real-world data is small compared to the generated data amount. Secondly, the training data does not require incorrect records as examples for outliers, in contrast to many state-of-the art outlier detection models which require labeled true positives. Thus, no additional data collection exceeding the requirements for the data generators is necessary. Thirdly, the evaluation enables a structural evaluation without aggregated KPIs, since the plausibility of every record is tested for sequences as well as combinations. In particular, the compassion using the weight matrices and Frobenius norm uses the results of a training over the full dataset and therefore includes the structural information about all records. Fourthly, differences between data generators are explained by comparing the weight matrices directly and thus, modified Hopfield neural networks are highly interpretable models. Therefore, they add explanatory value to the differences between synthetic and real-world data of "black-box" generators, such as the transformer-based approach.

However, the approach has two limitations. Firstly, the distance values of the distance matrices cannot be interpreted directly and thus, there exists no distance threshold where the real-world and synthetic data set are perceived as too different or the data generator as insufficient. The evaluation only compares different generators and selects the most suitable one from a collection but does not evaluate if any generator is suitable. Thus, the distance can only be interpreted in the context of other generators. However, optically close energy distributions (i.e., a small distance between the energy distributions using the Wasserstein metric) are a good indicator that the generated data is very similar. The decision whether the similarity

is sufficient must still be conducted on a case-by-case basis. In the here presented example, the energy distributions between the real-world data and transformer-based generated data are so close that the generator can be perceived as sufficient. However, this might not be the case in all use cases.

Secondly, the comparison is limited by separate analyses of combination and sequence information and the selection between two metrics. Only one criterion and metric are evaluated per distance matrix. In the given use case, the performance of the generators is comparable for all scenarios. However, this might not always be the case. If differences in the generators' performances between each evaluation occur, a case-by-case decision is required since no unifying metric exists. This decision requires domain knowledge whether combination or sequence information are more important for the specific use case. In addition, the selection of the metric must be assessed thoroughly considering the discussed differentiation.

Both limitations are mitigated if the validation is conducted carefully. The first limitation is addressed by evaluating the average distance between multiple subsets randomly drawn from the real-world dataset. This enables the evaluation of a threshold distance. Datasets within this threshold distance are then considered to be originating from the same dataset. Thus, it is possible to estimate a threshold under which synthetic data is considered as realistic. However, this threshold is use case dependent and not generalizable. The second limitation can be mitigated prior to the analysis by carefully considering domain knowledge on which metric and if sequences or combinations are more crucial for plausible synthetic datasets.

6 CONCLUSION

In summary, the proposed methodology proves itself suitable for a data generation and validation approach. First, the data generation models do not result in an unusual number of outliers. Second, the methodology can filter the small amount of outlier data successfully and enable a generation of more plausible datasets. Third, the validation procedure provides meaningful recommendations and is robust regarding the analysis of either using the machinery combinations or sequences, as well as robust regarding the selection of the metric, i.e., the Wasserstein metric comparing the energy distributions or the Frobenius norm comparing the individual weight matrices of the trained modified Hopfield neural networks. Therefore, the proposed method can be recommended for data generation approaches for machinery sequences in manufacturing applications. Potential applications are in the field of production planning, intralogistics, and scheduling.

The presented use case can be improved by future research with an evaluation using different product groups and product structures. In addition, the embedding of order arrival dates as a relevant data record in manufacturing use cases can additionally be considered in the data generation and validation to enhance the usefulness of the presented approach. Furthermore, future research should focus on a more detailed analysis of the capabilities of the modified Hopfield neural networks in other use cases. In particular, the robustness and explanatory power of the different metrics should be carefully evaluated. Further experiments are necessary to ensure the realism of the generated data. This can be achieved by comparing the synthetic data against more and divers real-world scenarios and by evaluating the usefulness of the results of simulations based on generated synthetic data, providing a direct measure of practical applicability and realism.

REFERENCES

Adolphy, S., H. Grosser, L. Kirsch, and R. Stark. 2015. "Method for Automated Structuring of Product Data and its Applications". *Procedia CIRP* 38:153–158.

Arinez, J. F., Q. Chang, R. X. Gao, C. Xu and J. Zhang. 2020. "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook". *Journal of Manufacturing Science and Engineering* 142(11):110804.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, *et al*. 2020. "Language Models are Few-Shot Learners". In *Advances in Neural Information Processing Systems*, Volume 33, 1877–1901.

Brush, S. G. 1967. "History of the Lenz-Ising Model". *Rev. Mod. Phys.* 39:883–893.

Fernandes, E. C., L. I. dos Santos, J. A. Camatti, L. Brown and M. Borsato. 2020. "Flexible Production Data Generator for Manufacturing Companies". *Procedia Manufacturing* 51:1478–1484.

Flamary, R., N. Courty, A. Gramfort, M. Alaya, A. Boisbunon, S. Chambon, *et al*. 2021. "POT: Python Optimal Transport". *Journal of Machine Learning Research* 22(78):1–8.

Hopfield, J. J. 1984. "Neurons with graded response have collective computational properties like those of two-state neurons.". *Proceedings of the National Academy of Sciences* 81(10):3088–3092.

Jaccard, P. 1901. "Distribution de la flore alpine dans le Bassin des Dranses et dans quelques régions voisines".

Koller, D. and N. Friedman. 2009. *Probabilistic graphical models: principles and techniques*. Cambridge: MIT press.

Krockert, M., M. Matthes, T. Munkelt, and S. Völker. 2021. "Generierung realitätsnaher Testdaten für die Simulation von Produktionen". In *19. ASIM Fachtagung Simulation in Produktion und Logistik 2021*, 565–574.

Libes, D., D. Lechevalier, and S. Jain. 2017. "Issues in synthetic data generation for advanced manufacturing". In *2017 IEEE International Conference on Big Data (Big Data)*, 1746–1754.

Mannino, M. and A. Abouzied. 2019. "Is this Real? Generating Synthetic Data that Looks Real". In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 549–561.

Matthes, M., O. Guhr, T. Munkelt, M. Krockert and S. Völker. 2023. "Maschinelles Lernen von Maschinenfolgen für den simulationsbasierten Test von Verfahren der Produktionsplanung und -steuerung". In *20. ASIM Fachtagung Simulation in Produktion und Logistik 2023*, 167–176.

Mourtzis, D. 2020. "Simulation in the design and operation of manufacturing systems: state of the art and new trends". *International Journal of Production Research* 58(7):1927–1949.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al*. 2011. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12:2825–2830.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al*. 2019. "Language models are unsupervised multitask learners". *OpenAI blog* 1(8):9.

Rubner, Y., C. Tomasi, and L. J. Guibas. 2000. "The Earth Mover's Distance as a Metric for Image Retrieval". *International Journal of Computer Vision* 40(2):99–121.

Schuh, G. 2007. *Produktionsplanung und -steuerung: Grundlagen, Gestaltung Und Konzepte*. Dordrecht: Springer.

Spoor, J. M. and J. Weber. 2024. "Evaluation of process planning in manufacturing by a neural network based on an energy definition of hopfield nets". *Journal of Intelligent Manufacturing* 35(6):2625–2643.

Tyugu, E. and E. K. Tyugu. 2007. *Algorithms and architectures of artificial intelligence*. Amsterdam: IOS Press.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al*. 2017. "Attention is All you Need". In *Advances in Neural Information Processing Systems*, Volume 30.

Völker, S., T. Döring, and T. Munkelt. 2001. "The Generation of Large Test Data for the Empirical Analysis of Heuristic Procedures for Production Planning and Control". In *Operations Research Proceedings*, 266–270.

Wuennenberg, M., K. Muehlbauer, J. Fottner, and S. Meissner. 2023. "Towards predictive analytics in internal logistics – An approach for the data-driven determination of key performance indicators". *CIRP Journal of Manufacturing Science and Technology* 44:116–125.

AUTHOR BIOGRAPHIES

JAN MICHAEL SPOOR Jan Michael Spoor pursued his doctoral degree at the Institute for Information Management in Engineering at the Karlsruhe Institute of Technology in cooperation with Mercedes-Benz. His research topics are anomaly detection, Digital Twins, and AI support systems. Currently, he works as an IT consultant at BCG Platinion. Prior, he worked at Homburg & Partner as a consultant specialized in B2B market strategy, sales, and pricing. His email address is jan.spoor@kit.edu.

MARVIN MATTHES is an AI Research Scientist for Production Planning and Control. Holding a Master's degree in Business Informatics, he specializes in causal analysis techniques to optimize production workflows in production planning and concepts of AI-driven production. His career has included roles as a Developer Consultant at GISA GmbH, with expertise in IT infrastructure projects and software development. His email address is marvin.matthes@htw-dresden.de.

MARTIN KROCKERT is a post doc at the Dresden University of Applied Sciences. His research focuses on distributed artificial intelligence and self-organizing systems. He accumulated industrial experience as a business process developer at Howden Group, where he led system integration and information workflow automation projects. He received his doctoral degree of engineering in software technology at the Technical University Dresden. His email address is martin.krockert@htw-dresden.de.

JENS WEBER Jens Weber is professor of mechanical engineering and Head of Mechatronics Trinational at Baden-Wuerttemberg Cooperative State University. He received his doctorate at the Heinz Nixdorf Institute Paderborn at the Chair of Business Computing, especially CIM. He then worked as IT project leader for the production engineering department of the digital factory and tool management of Mercedes-Benz AG. His email address is weberj@dhbw-loerrach.de.