




Article

Towards an Assembly Support System with Dynamic Bayesian Network

Stefan-Alexandru Precup¹, Arpad Gellert^{1,*}, Alexandru Matei¹, Maria Gita^{2,3}
and Constantin-Bala Zamfirescu¹

¹ Computer Science and Electrical Engineering Department, Lucian Blaga University of Sibiu, 550025 Sibiu, Romania; stefan.precup@ulbsibiu.ro (S.-A.P.); alex.matei@ulbsibiu.ro (A.M.); constantin.zamfirescu@ulbsibiu.ro (C.-B.Z.)

² Department of Industrial Engineering and Management, Lucian Blaga University of Sibiu, 550025 Sibiu, Romania; maria.gita@ulbsibiu.ro

³ IFM Prover, 557085 Sibiu, Romania

* Correspondence: arpad.gellert@ulbsibiu.ro

Abstract: Due to the new technological advancements and the adoption of Industry 4.0 concepts, the manufacturing industry is now, more than ever, in a continuous transformation. This work analyzes the possibility of using dynamic Bayesian networks to predict the next assembly steps within an assembly assistance training system. The goal is to develop a support system to assist the human workers in their manufacturing activities. The evaluations were performed on a dataset collected from an experiment involving students. The experimental results show that dynamic Bayesian networks are appropriate for such a purpose, since their prediction accuracy was among the highest on new patterns. Our dynamic Bayesian network implementation can accurately recommend the next assembly step in 50% of the cases, but to the detriment of the prediction rate.

Keywords: assembly assistance system; dynamic Bayesian network



Citation: Precup, S.-A.; Gellert, A.; Matei, A.; Gita, M.; Zamfirescu, C.-B. Towards an Assembly Support System with Dynamic Bayesian Network. *Appl. Sci.* **2022**, *12*, 985. <https://doi.org/10.3390/app12030985>

Academic Editors: Stanislao Patalano, Antonio Lanzotti, Bruno Siciliano and Luigi Villani

Received: 9 November 2021

Accepted: 17 January 2022

Published: 19 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Industry 4.0 concepts are surprising the manufacturing industry by requiring continuous transformations due to technological advancement in internet technologies, sensors, and data processing hardware. On the other side, the customers are pressuring the production lines with highly customized, small batch orders. These recent requirements shift the paradigm from a rigid, static factory to more flexible, automated, and intelligent production lines that need to dynamically update the production plan based on customer orders and other events—predicted or unexpected. Even though repetitive or dangerous actions and operations are becoming automated, the human workers are not removed entirely from the manufacturing process. To keep the human and the manual assembly operations in line with the digitalization of the industry, new concepts, such as Operator 4.0 [1], Healthy Operator 4.0 [2], or Assembly 4.0 [3,4], emerged. Operators are gaining new responsibilities and are helped in the process by continuous training or by receiving real-time support with the help of devices such as smart glasses and augmented reality technology [5–7]. The manual assembly processes are modeled to be flexible and are monitored with the help of motion capture technologies [8,9]. All the sensors used for this generate a lot of industrial data that is analyzed with the help of machine learning and artificial intelligence algorithms so they can be further improved and optimized or to identify unusual states of the processes.

In this work, we present dynamic Bayesian networks (DBN) applied to predict the next assembly step during an assisted manual manufacturing process. The proposed approach will become a part of a prediction module of a larger control system used on a physical assembly training station that can adapt the instruction sequence and content based on

what the operator is doing at that moment. The control system, described in [10] as part of the digital twin of the assembly training station, consists of multiple sensors that are used to obtain the current state of the assembly, several modules used for prediction and adaptation, together with a large visual interface where the operator sees the output of the control system. Depending on the use case, the next assembly step prediction can be used either to anticipate what the operator will do and prepare in advance for his actions or as hints that the operator can use and follow.

The DBN relies also on Markov chains for establishing its initial probabilities. The network uses as input data the current assembly state of the product and static characteristics of the human worker, such as the height, the gender, if it wears glasses or not, and the sleep quality in the previous night. The workers' assembly preferences were highly influenced by their gender (e.g., first picked component based on its color). Their height affected the order of picked components, shorter workers picking the components located closer to them first. The sleep quality affects their concentration capabilities, reflected in the number of mistakes made during the assembly. Wearing eyeglasses also influenced the assembly process.

Additionally, we evaluate the usefulness of applying some dynamic human characteristics, such as the mood in each stage of the assembly process. Our DBN-based assembly prediction method is evaluated on a dataset collected during an experiment involving 68 students, described in [11] and [12]. We will also compare it with our previous work on next assembly step predictors.

The rest of this paper has the following structure: Section 2 presents the related work, Section 3 describes the DBN-based next assembly step prediction, Section 4 discusses the results, and Section 5 concludes the paper.

2. Related Work

Cyber-physical systems produce major changes in entire industries, especially in the factory area, where there is an ever-increasing discussion about the role of operators in the production processes. In [13], the authors identified an increased demand in highly customizable smart products. Due to the overhead costs, the use of a human workforce is preferred. They are developing a scalable assistance module for the assembly process with a big focus on the ergonomics of the workplace.

In [14], the authors wrote about how the assistance systems increase productivity in assembly processes, and they formulated the principles of designing assembly assistance systems. They stated that it is unclear what problems might appear in the representation of information in manual assemblies or how these problems should be solved using assistive systems. By using two example cases, the authors could identify such problems. The authors split the identified problems into five categories of information representation problems: scarcity of the input provided to the work system, irrelevancy of data, outdated information, and lack of process orientation, incompatibility of information representation with human interpretation. From all these problems, we partially encountered data irrelevancy. We addressed this problem by eliminating irrelevant data.

The authors of [15] state that the manufacturing trends are heading towards high-variant and small quantity assembly of products. Companies started using cognitive assistance systems with the aim of increasing the quality and efficiency while decreasing the cognitive load of the workers. They state that the workers' lack of acceptance towards assistive systems decreases their benefits. An approach for an assembly workstation with multiple software and hardware components was proposed. Similarly, our system has various software and hardware components, the main differences being the presence of the next assembly step suggestion software component and the usage of a large touchscreen instead of a projector for visual communication with the worker.

In [16], the authors analyzed the technology that an assistive system should encompass. They started by analyzing the existing assembly assistance systems and how they can optimize those systems. Experts were interviewed to determine what functionalities

an assembly assistance system should have. Several methods were proposed for implementation, such as intelligent image processing, deep learning algorithms, gamification, or augmented reality. In our work, due to the modular nature of our customizable tablet and a high number of assembling possibilities, gamification represents a core principle.

Assembly support systems can be found in the literature with different implementation complexities and enabled by different technologies. The simplest ones allow the operators to visualize the instructions. The operators move to the next step when they consider that the current step is finished and carried out correctly. Examples include traditional assembly manuals or new approaches that use AR technologies [17,18] that superimpose the information over the real object. Advanced assembly support systems can also extract information about the current step in the assembly process. This is achieved by recognizing the current user action, fusing information from different sensors, such as electromyography sensors, inertial measurement unit, and depth RGB camera [19]. Another way is to recognize the current assembly state of the product using a simple RGB camera, as in [20], for example.

More complex assembly support systems use the information about the current state of the system to predict future human actions [20–22]. The use case from [20] predicts the next assembly state to detect faults and mistakes in a predefined and ordered assembly process. This is achieved using an end-to-end neural network that takes as input only images of the assembled product without any other information about the operator. The authors of [21] describe a human robot collaboration system that anticipates the parts needed by the operator and delivers them into three separate containers. The system predicts the points in time when the operator will need a part and will plan accordingly that the operator always has the three most likely needed parts available. The system is evaluated by the total assembly process execution time and the time the operator waits for the robot to deliver the needed parts. In [22], a variable length Markov model is used to predict the next operator actions using a temporal context recognized by a bi-stream convolutional neural network. The temporal context consists of the previous and current actions of the operator, without other additional features. Some drawbacks of [22] are the small number of experiment subjects and that the assembly is heavily restricted, with only two states in which the operator can choose his course of action. In contrast, in our work, we use DBN for assembly step prediction and our product has much more flexibility, allowing freedom of choice at any of the assembly states.

DBNs have major applications in different fields: from gene sequence modeling in [23,24] and crash prediction based on traffic speed in [25] to exchange rate predictability in [26]. In [27], the authors propose an approach for activity recognition based on DBN. They divided the features that describe the object motions in two classes: global and local, which are two different spatial scales. The global features describe the movement of the object at a large scale and the relations between the objects or the environment. Meanwhile, the local features represent the movement of the object of interest. The proposed DBN structure has a state duration that models the human interacting activities. Furthermore, the authors present the effectiveness of their approach with an experiment.

The authors of [28] present an approach to measure network security using DBN. They stated that the security metrics measured individual vulnerabilities without consideration towards their combined effect. They propose a DBN model that aims to incorporate the temporal factors, such as the availability of patches or exploit codes. Potential applications of the DBN-based assembly prediction method are analyzed. Furthermore, they present how the DBN can be obtained from attack graphs and how it can be used to analyze the networks' security aspects. In our model, the time is incorporated through the sequentiality of the assembly steps.

In [29], the authors use DBN for web search ranking. They state that the page position affects the number of times the page is clicked: the lower the position of the page, the less likely the user is to click on that page. This is called position bias. They propose a DBN

that aims to make an unbiased estimation of the relevance based on click logs. Their model has outperformed other click models in both clickthrough rate and relevance.

In our previous works, we evaluated different next assembly step prediction methods. In [30], we analyzed two-level context-based predictors, which use the assembly context built up in the first level to select the corresponding pattern of the prediction table from the second level, whose associated next state will be the predicted one. The Markov predictor, presented in [31], improves the two-level predictor by storing for each pattern, beside the next states, the frequency of their apparition, thus predicting the state with the highest frequency. The Markov predictor was enhanced in [11] with a padding mechanism. In [12], we applied a prediction by partial matching algorithm that internally uses the Markov predictor presented in [11]. Furthermore, in [32], we implemented a long short-term memory recurrent neural network for the prediction of next assembly steps. The results of the prediction methods mentioned above will be presented in Section 4 comparatively with the results of the proposed DBN-based predictor. The DBN proved to be the most accurate model.

3. Next Assembly Step Prediction through Dynamic Bayesian Network

This section briefly presents the target product used in this work and describes the DBN as a prediction model, including its implementation in our assembly assistance system.

3.1. The Target Product

Our assembly assistance system was presented in detail in [11]. It retrieves information about the worker and the assembled product and can provide support for the next assembly steps.

As in our previous works [11,12,30–32], the manufactured product (visible in Figure 1) is a modular tablet built out of 8 components: a mainboard, a screen, and six modules. A key characteristic of this product is that the assembly process is very flexible with no dependencies between the steps. This allows the operator full freedom to assemble the product. The mainboard is the component on which all the other components will be mounted. There are three types of modules: speaker modules (white pieces), flashlight modules (purple pieces), and battery modules (blue pieces). Two of each type are used in our product for a total of six modules. We described how we encoded the assembly state of the tablet in [12,30].



Figure 1. The assembled product from front, back, and perspective.

3.2. The DBN as a Prediction Model

A Bayesian network, known also as a belief or causal network, is a probabilistic graphical method that models variables together with their conditional dependencies using a directed acyclic graph (DAG). These networks are ideal for predicting the likelihood of known factors that contributed to the occurrence of an event.

If we consider the variables A , B , C , D , we can factor their joint probability $P(A, B, C, D)$ as a product of conditional probabilities, such that:

$$P(A, B, C, D) = P(A) \cdot P(B|A) \cdot P(C|A, B) \cdot P(D|A, B, C) \quad (1)$$

The factorization of variables A, B, C, D in Equation (1) does not provide any relevant information related to the joint probability distribution because each variable can depend on every other variable [33].

$$P(A, B, C, D) = P(A) \cdot P(B) \cdot P(C|A) \cdot P(D|B, C) \tag{2}$$

If we are considering the factorization in Equation (2), some conditional independent variables can be observed [33]. According to the theory of probability, two random variables (events) X and Y are conditionally independent given a third event Z if, knowing that Z will occur, event Y will not be influenced by the occurrence of event X and event X will not be influenced by the occurrence of event Y . Mathematically explained, X is independent of Y given Z if $P(X, Y|Z) = P(X|Z) \cdot P(Y|Z)$. From the factorization (Equation (2)), we can show that, given the events B and C , the events A and D are independent of each other.

$$P(A, D|B, C) = \frac{P(A, B, C, D)}{P(B, C)} = \frac{P(A) \cdot P(C|A) \cdot P(D|B, C)}{P(C)} = P(A|C) \cdot P(D|B, C) \tag{3}$$

A Bayesian network can be used to represent the factorization of the joint distribution and, for each random variable, there is a node associated with it in the network. A directed edge is drawn from a node X to another node Y , if Y is conditioned on X . Figure 2 is a representation of the above factorization (Equation (2)).

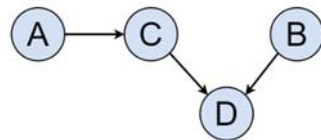


Figure 2. A directed acyclic graph (DAG) representing the factorization (Equation (2)).

Usually, a Bayesian network is built based on existing knowledge about the conditional independence of the variables and a dataset of observations. From the used dataset, we were able to identify variables that might influence the assembly steps.

In our case, we have five independent variables that were identified to influence the assembly state. Figure 3 presents our five variables and their conditional dependencies. Node H represents the height of the user, node G its gender, node S its sleep quality, node E if it wears eyeglasses, and node M_t the mood. All these five nodes represent binary variables that, for height, represent tall/small; for gender, male/female; for sleep quality, good/bad; for wearing glasses, true/false; and, for the mood, positive/negative. Node A_t is the assembly state of the product at time t , which is directly dependent on the other five variables.

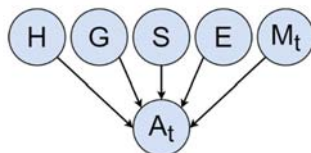


Figure 3. A Bayesian network with the five variables that influence the next assembly step.

This model is not useful for us to describe the assembly process due to the temporal nature of the assembly. Figure 4 represents a Bayesian network that considers the evolution in time of both the user’s mood and the assembly state of the tablet. Thus, the current assembly state directly depends on the human characteristics and on the previous assembly state, respectively.

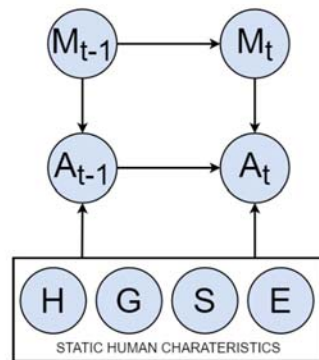


Figure 4. The proposed DBN architecture with the evolution of the mood and the assembly state of the tablet.

The Bayesian networks that can model the sequences, considering the time factor and the evolution of variables through it, are known as DBN and are also called temporal Bayesian networks.

A DBN connects variables to each other over adjacent time steps. The DBN can also be considered a two-time-slice Bayesian network (2TBN) since, at any given time, a variable’s value can be computed using the internal regressors and the actual variable’s value at time $t - 1$.

In the modelling of time series, the values of variables are observed at different time steps. Since time can move in only one direction (forward), the design of these networks is simplified and the directed edges should follow the direction of time (forward). If we consider a sequence of data $\{X_1, X_2, X_3, \dots, X_t\}$, with X_t representing the value of the variable X at time t , then the simplest model is actually a Markov model of the first order. The probability of that sequence is:

$$P(X_1, X_2, \dots, X_t) = P(X_1) \cdot P(X_2|X_1) \cdot \dots \cdot P(X_t|X_{t-1}) \tag{4}$$

The graphical representation of the temporal Bayesian network from Equation (4) is illustrated in Figure 5.

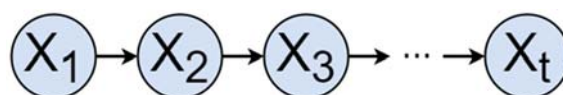


Figure 5. A temporal Bayesian network representation of a Markov process of order 1.

The proposed DBN, envisioned in Figure 4, has been implemented using the pgmpy library developed by Ankan Ankur and Panda Abinash [34]. We are using the five aforementioned user characteristics and the current assembly state in order to predict the next assembly state.

4. Experimental Results

For the experimental results, we are using the same dataset obtained from an experiment with 68 participants, presented in [11,12]. Briefly described, given two images of the target product, they had to assemble the components to get to the final product presented in Figure 1. The position of the components was identical for all the participants and they could pick at any time any component in order to complete the assembly. The assembly sequences were encoded and used as inputs for our models. The participants also filled in a questionnaire. There were questions regarding height, age, gender, dominant hand, and if the participants were eyeglass wearers. Other questions were for self-assessment: “were you hungry during the experiment?”, “do you have any prior experience in product assembly?”, “what was your stress level before the experiment?”, “are you under the

influence of any drugs that might influence your level of concentration?”, and “how would you describe the sleep quality of the previous night?” During the experiment, we also collected the participants’ mood.

There are three metrics of interest: accuracy, prediction rate, and coverage:

$$Accuracy = \frac{Correct\ predictions}{Predictions\ made} \tag{5}$$

$$Prediction\ Rate = \frac{Predictions\ made}{Dataset\ size} \tag{6}$$

$$Coverage = \frac{Correct\ predictions}{Dataset\ size} \tag{7}$$

The accuracy (Equation (5)) measures how well the DBN-based model predicts, the prediction rate (Equation (6)) indicates how many times the model can actually predict, and the coverage (Equation (7)), which can be considered the most important metric, provides information regarding the rate of correct predictions from the whole testing dataset.

In our previous works, the algorithms were validated using two evaluation methods. One was used to determine the capability of the evaluated methods to learn existing scenarios (using the whole dataset in both training and testing phases). The other evaluation method allowed observation of how well the methods will adapt to new scenarios (the correct assemblies from the first three quarters of the dataset were used for training and the last quarter for evaluation). Although the data provided have a diversity of assemblies, to mitigate the selection bias that the former evaluation methods might have introduced, a cross-validation method has been considered in the current work.

Figure 6 describes the flow of our experiment. Using the existing preprocessed dataset, we create two datasets: one that takes into account the mood variable of the user and one that does not. We evaluate the DBN model and we compare it to our previous PPM, Markov, and LSTM models. The training of the models is carried out using the k-fold cross-validation method. This method consists of splitting the dataset in k equal subsamples. Out of these k subsamples, one subsample is used for the evaluation of the model, while the remaining $k - 1$ are used for the training of the model. Afterwards, this cross-validation evaluation method is repeated k times, with all the subsamples being used only once as an evaluation sample.

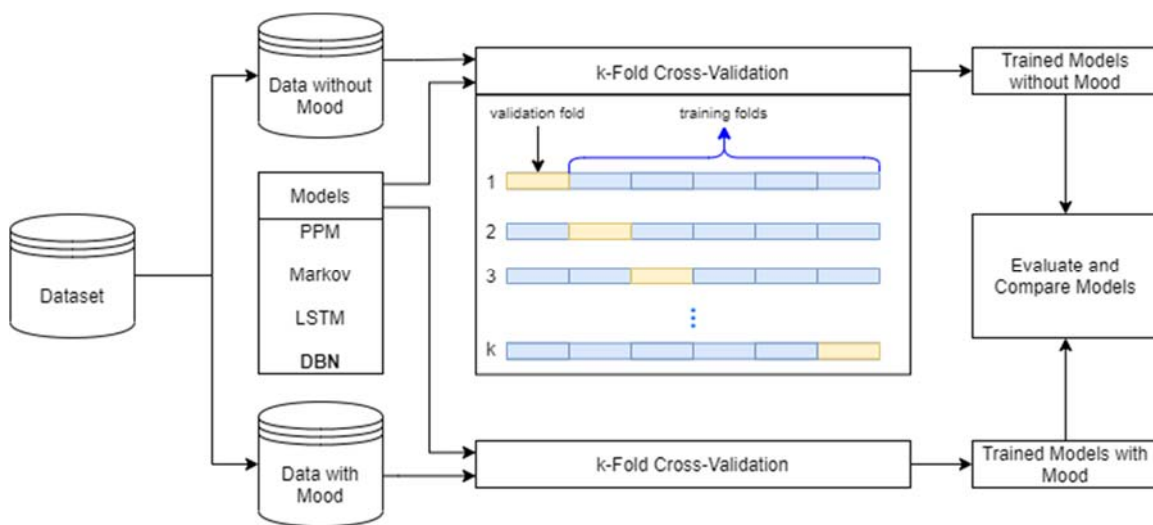


Figure 6. The flow of the experiment.

We ranged the k number of subsamples from 2 to 6. We did not select a k greater than 6 as the accuracy and coverage start to decrease due to overfitting of the data. The graphics

that will be further presented contain the average of each k-fold run. Furthermore, the overall average is also available.

To determine the features that are significant for our prediction model, we computed the F -value and the p -value on the dataset (Table 1). The goal was to determine which features of the human worker led to the most correct assemblies. We selected the features having a p -value less than 0.1, because a p -value greater than 0.1 indicates insufficient evidence [35]. Thus, we chose to be used as additional input data in the prediction models and binarized the following features: gender (male/female), sleep quality (good/bad), eyeglass wearer (yes/no), and height (tall/small—with the threshold of 174 cm, the average height of the participants).

Table 1. Feature significance.

Feature	F -Value	p -Value
Assembly Experience	0.00079	0.97762
Age	0.10553	0.74631
Stress level before the assembly	0.36950	0.54535
Hungry	0.55439	0.45917
Under influence of medication	0.69261	0.40827
Preferred hand	2.40527	0.12570
Gender	2.86426	0.09528
Sleep quality	2.87701	0.09456
Eyeglass wearer	3.99500	0.04975
Height	6.98954	0.01023

Next, the proposed DBN predictor is compared with the order 1 Markov model presented in [2], the long short-term memory (LSTM) recurrent neural network presented in [32], and the order 3 prediction by partial matching (PPM) with neighbor exploration presented in [12].

Figure 7 presents the prediction rate of the selected methods. As can be observed, the LSTM network has the highest prediction rate across all runs, averaging 96%. Compared with LSTM, PPM predicts 23% less and DBN 46% less. Even though LSTM has a high prediction rate, as can be observed in Figure 8, the accuracy of its predictions is lower compared with the other methods, having an accuracy of only 28%. Both the PPM and Markov models have a very high accuracy of 49%. The DBN has the highest prediction accuracy of 50%.

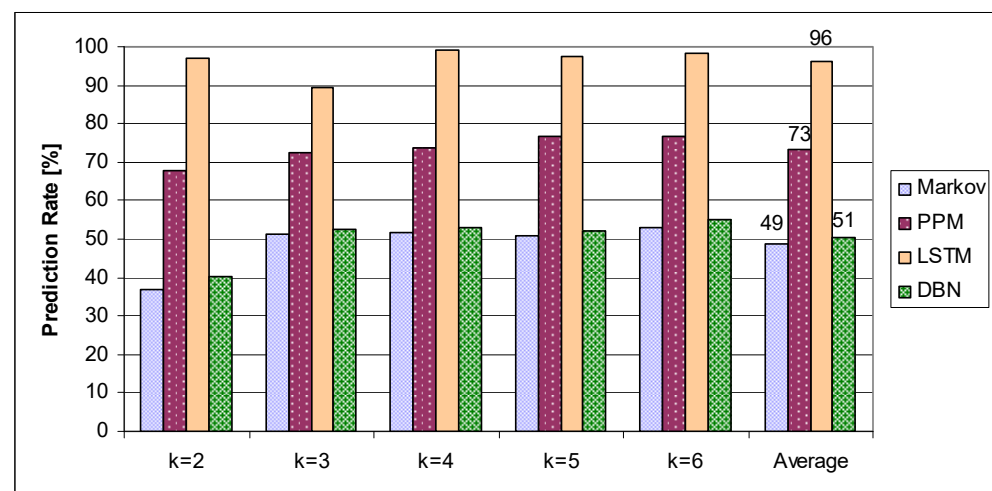


Figure 7. Prediction rate without the mood variable through k-fold cross-validation.

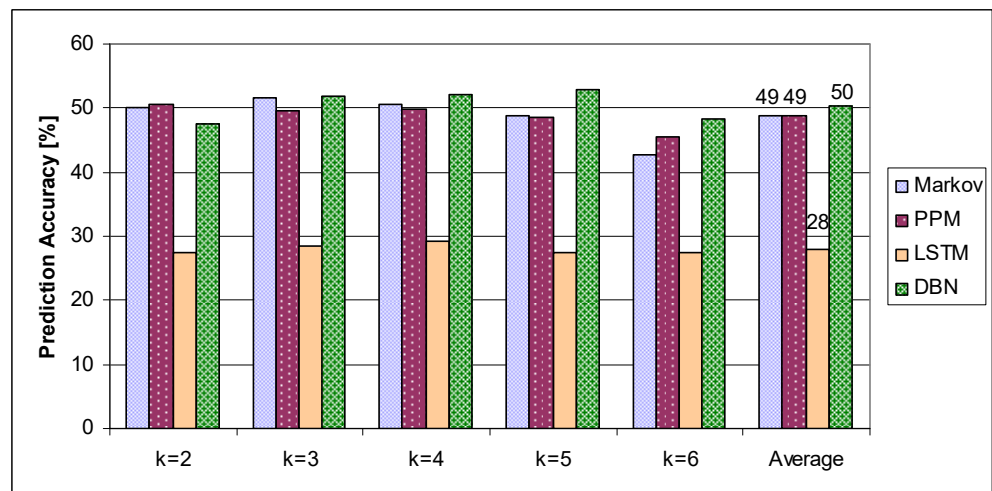


Figure 8. Prediction accuracy without the mood variable through k-fold cross-validation.

The coverage is strongly correlated with the prediction rate (see Figure 9). Surprisingly, the best coverage of 36% is obtained by using the PPM implementation. The LSTM has a coverage of only 27%, despite its high prediction rate. The DBN has a coverage of 26%, slightly higher than the Markov model’s 24%.

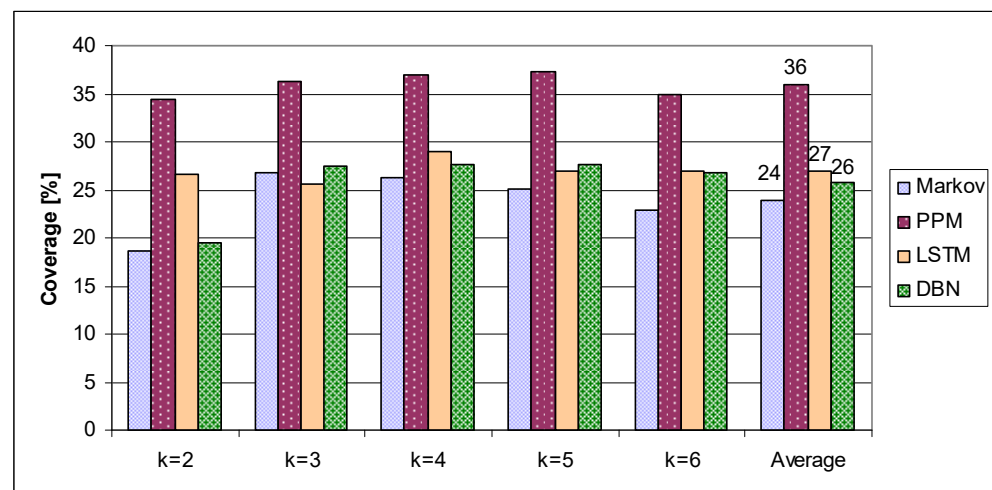


Figure 9. Coverage without the mood variable through k-fold cross-validation.

Besides the four core human characteristics (height, gender, sleep quality, and if the worker wears glasses), all the above methods have been further enhanced to take into consideration the mood of the user in the prediction process, thus obtaining a better representation of the human worker. The mood of the user is recorded with the pretrained emotion recognition model from Intel’s OpenVino toolkit [36]. It is able to identify five states (neutral, happy, sad, surprise, and anger), which were binarized into either a positive or negative mood of the user.

In Figure 10, a decrease in prediction rate across all the prediction methods was observed, except for LSTM, which has a slightly higher prediction rate of 97%. The DBN has a decrease of 3% in the predictions made, while PPM predicts 14% less steps. The Markov predictor is the most influenced by the addition of the mood variable, being able to predict only 33% of the assembly steps, which is a decrease of 16% compared to the implementation without the mood variable.

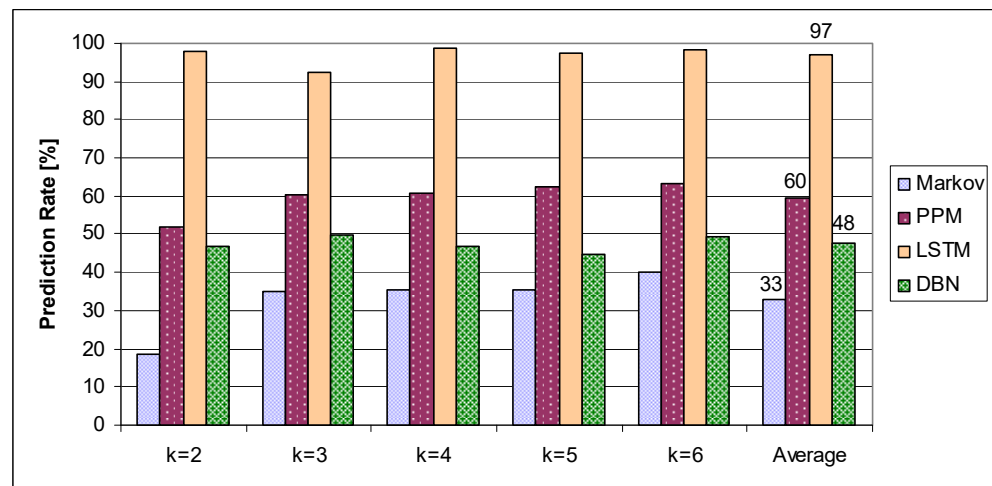


Figure 10. Prediction rate with the mood variable through k-fold cross-validation.

A decrease in accuracy can be observed for all the methods in Figure 11. While, for the PPM and LSTM, the decrease is small, of only 3% and 4%, respectively, the Markov and DBN predictors seem to lose considerable percentages in their prediction accuracy. The Markov predictor has a 10% decrease compared to the one that does not take the mood into consideration, thus having an effective prediction accuracy of 39%. The DBN seems to take a big impact to its prediction accuracy, with a negative change of 14%, thus obtaining a prediction accuracy of 37%, being very close to that obtained by using the Markov model.

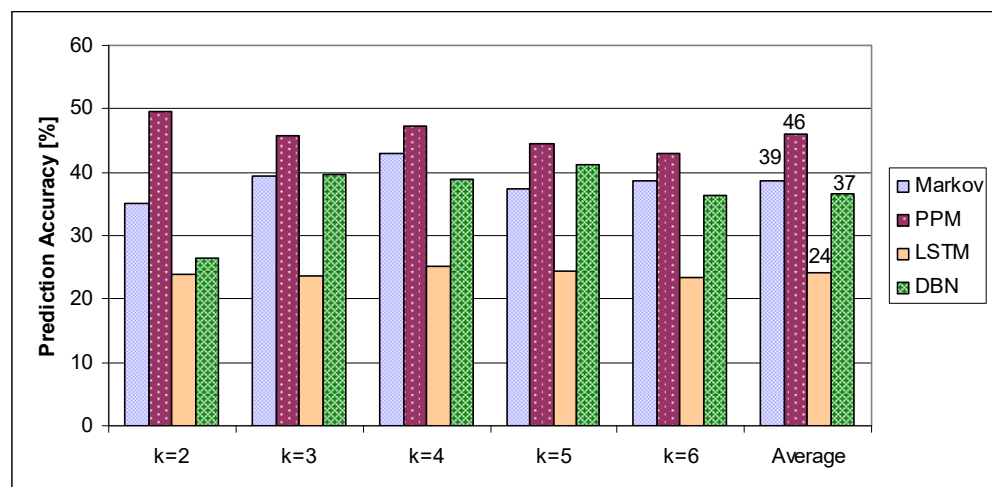


Figure 11. Prediction accuracy with the mood variable through k-fold cross-validation.

Figure 12 presents the coverage of the evaluated methods, including the mood variable. As in the case of the prediction accuracy, the coverage is decreased for all the methods. The PPM has the highest coverage of all the predictors at 27%. The DBN has a coverage of 17%, a decrease of 9% compared to the implementation without the mood variable. Consequently, the mood variable does not increase the performance and, thus, the optimal DBN relies only on the current assembly state and the initial four human characteristics—height, gender, sleep quality, and if the worker wears glasses—to predict the next assembly state. Without considering the mood variable, even though the PPM has a higher coverage compared to the DBN, if we want to determine with the highest accuracy the next move of the worker, then the use of the DBN predictor is preferred, since it has the highest prediction accuracy among all the presented methods. We use the accuracy metric as the determining factor for choosing a predictor, as we wish to give the worker as few false predictions as possible,

even though that will mean, in some cases, the predictor might not be able to provide one due to its low prediction rate.

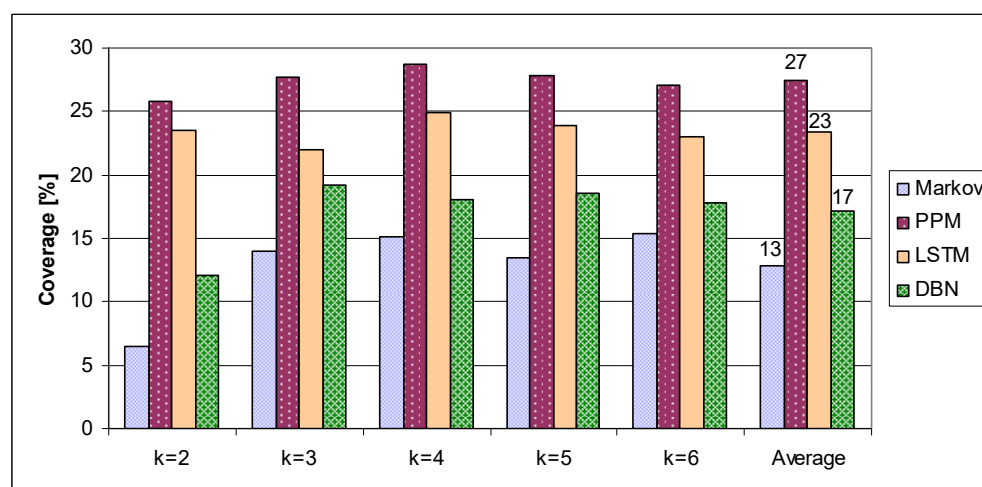


Figure 12. Coverage with the mood variable trough k-fold validation.

Overall, by looking at Figures 7–12, due to the use of the k-fold cross-validation method, we can observe interesting facts about the compared predictors. The models reach peak performance at different values of k : the PPM model reaches the highest coverage for $k = 5$, while the DBN reaches it at $k = 3$, showing that the DBN model can converge optimally with a smaller size of training dataset. The LSTM model does not seem to be affected that much by the values of k , especially $k = 2$, where the metrics of the other methods drop significantly, especially in the case of the models where the mood variable is included. The downside of the LSTM is that the metrics of accuracy and coverage are very low and they do not improve by increasing the training size. For $k = 6$, the accuracy and the coverage decrease for the Markov, PPM, and DBN models, indicating an overfitting problem.

5. Conclusions

In this work, the DBN was studied as an assembly step predictor. The utilization of the DBN in an assembly assistance station for the guidance of the workers is an original contribution. The DBN was optimally configured and compared with other existing prediction methods in terms of prediction accuracy, prediction rate, and coverage. The DBN-based assembly prediction method was validated on a dataset composed of the assemblies and the human characteristics of 68 trainees and will be integrated into the control system of an existing manual assembly training station. The evaluation results have shown that the DBN is able to provide the highest prediction accuracy, with 50% of its predictions being correct, at the expense of a lower prediction rate (51%) and coverage (26%). As we are interested in including the best performing assembly modeling method into our human-oriented assembly assistance system, we intend to further study the applicability of the A* algorithm and hidden Markov models. Finally, the method with the best preliminary results will be validated in an industrial environment. In this case, we expect a simplification in the data gathering of user characteristics, human workers having their own profile with no need for questionnaires. The factory workers must be familiarized with the assembly assistance system before using it in the manufacturing process. Moreover, an industrial context allows a large amount of real-world data to be continuously collected that will improve the prediction accuracy by continuous learning for a certain product. For each new product, a new encoding is necessary. This implies that the proposed method must be applied separately for each product. Any modification in feature significance due to the larger dataset will not require changes to any of the above-mentioned algorithms.

Author Contributions: Conceptualization, A.G. and C.-B.Z.; methodology, A.G. and S.-A.P.; software, S.-A.P.; validation, S.-A.P.; formal analysis, A.G., S.-A.P. and C.-B.Z.; investigation, A.G. and S.-A.P.; resources, C.-B.Z.; data curation, S.-A.P. and A.G.; writing—original draft preparation, S.-A.P., A.G., A.M., C.-B.Z. and M.G.; writing—review and editing, S.-A.P., A.G., A.M., C.-B.Z. and M.G.; visualization, S.-A.P. and A.G.; supervision, A.G. and C.-B.Z.; project administration, C.-B.Z.; funding acquisition, C.-B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a Hasso Plattner Excellence Research Grant (LBUS-HPI-ERG-2020-03), financed by the Knowledge Transfer Center of the Lucian Blaga University of Sibiu.

Institutional Review Board Statement: All the experiments presented and used in this study were approved by the Research Ethics Committee of Lucian Blaga University of Sibiu (No. 3, on 9 April 2020).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ruppert, T.; Jaskó, S.; Holczinger, T.; Abonyi, J. Enabling Technologies for Operator 4.0: A Survey. *Appl. Sci.* **2018**, *8*, 1650. [[CrossRef](#)]
2. Sun, S.; Zheng, X.; Gong, B.; Paredes, J.G.; Ordieres-Meré, J. Healthy Operator 4.0: A Human Cyber-Physical System Architecture for Smart Workplaces. *Sensors* **2020**, *20*, 2011. [[CrossRef](#)]
3. Bortolini, M.; Ferrari, E.; Gamberi, M.; Pilati, F.; Faccio, M. Assembly system design in the Industry 4.0 era: A general framework. *IFAC-PapersOnLine* **2017**, *50*, 5700–5705. [[CrossRef](#)]
4. Miqueo, A.; Torralba, M.; Yagüe-Fabra, J.A. Lean Manual Assembly 4.0: A Systematic Review. *Appl. Sci.* **2020**, *10*, 8555. [[CrossRef](#)]
5. Longo, F.; Nicoletti, L.; Padovano, A. Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Comput. Ind. Eng.* **2017**, *113*, 144–159. [[CrossRef](#)]
6. Danielsson, O.; Holm, M.; Syberfeldt, A. Augmented reality smart glasses in industrial assembly: Current status and future challenges. *J. Ind. Inf. Integr.* **2020**, *20*, 100175. [[CrossRef](#)]
7. Santi, G.; Ceruti, A.; Liverani, A.; Osti, F. Augmented Reality in Industry 4.0 and Future Innovation Programs. *Technologies* **2021**, *9*, 33. [[CrossRef](#)]
8. Faccio, M.; Ferrari, E.; Gamberi, M.; Pilati, F. Human Factor Analyser for work measurement of manual manufacturing and assembly processes. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 861–877. [[CrossRef](#)]
9. Ling, S.; Guo, D.; Zhang, T.; Rong, Y.; Huang, G.Q. Computer Vision-enabled Human-Cyber-Physical Workstations Collaboration for Reconfigurable Assembly System. *Procedia Manuf.* **2020**, *51*, 565–570. [[CrossRef](#)]
10. Matei, A.; Tocu, N.-A.; Zamfirescu, C.-B.; Gellert, A.; Neghina, M. Engineering a Digital Twin for Manual Assembling. In *Part IV. Leveraging Applications of Formal Methods, Verification and Validation: Tools and Trends, Proceedings of the 9th International Symposium on Leveraging Applications of Formal Methods, ISOFA 2020, Rhodes, Greece, 20–30 October 2020*; Tiziana, M., Bernhard, S., Eds.; Springer: Cham, Switzerland, 2021; Volume 12479. [[CrossRef](#)]
11. Gellert, A.; Precup, S.-A.; Pirvu, B.-C.; Zamfirescu, C.-B. Prediction-Based Assembly Assistance System. In *Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation, Vienna, Austria, 8–11 September 2020*; pp. 1065–1068. [[CrossRef](#)]
12. Gellert, A.; Precup, S.-A.; Pirvu, B.-C.; Fiore, U.; Zamfirescu, C.-B.; Palmieri, F. An Empirical Evaluation of Prediction by Partial Matching in Assembly Assistance Systems. *Appl. Sci.* **2021**, *11*, 3278. [[CrossRef](#)]
13. Nguyen, T.D.; McFarland, R.; Kleinsorge, M.; Krüger, J.; Seliger, G. Adaptive Qualification and Assistance Modules for Manual Assembly Workplaces. *Procedia CIRP* **2015**, *26*, 115–120. [[CrossRef](#)]
14. Hinrichsen, S.; Bendzioch, S. How Digital Assistance Systems Improve Work Productivity in Assembly. In *Advances in Human Factors and Systems Interaction*; Nunes, I., Ed.; Springer: Cham, Switzerland, 2018; Volume 781, pp. 332–342. [[CrossRef](#)]
15. Sochor, R.; Kraus, L.; Merkel, L.; Braunreuther, S.; Reinhart, G. Approach to Increase Worker Acceptance of Cognitive Assistance Systems in Manual Assembly. *Procedia CIRP* **2019**, *81*, 926–931. [[CrossRef](#)]
16. Petzoldt, C.; Keiser, D.; Beinke, T.; Freitag, M. Functionalities and Implementation of Future Informational Assistance Systems for Manual Assembly. In *Subject-Oriented Business Process Management. The Digital Workplace—Nucleus of Transformation. S-BPM ONE 2020. Communications in Computer and Information Science*; Freitag, M., Kinra, A., Kotzab, H., Kreowski, H.J., Thoben, K.D., Eds.; Springer: Cham, Switzerland, 2020; Volume 1278, pp. 88–109. [[CrossRef](#)]
17. Blankemeyer, S.; Wiemann, R.; Raatz, A. Intuitive Assembly Support System Using Augmented Reality. In *Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter*; Schüppstuhl, T., Tracht, K., Franke, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2018. [[CrossRef](#)]

18. Wang, Y.; Zhang, S.; Wan, B.; He, W.; Bai, X. Point cloud and visual feature-based tracking method for an augmented reality-aided mechanical assembly system. *Int. J. Adv. Manuf. Technol.* **2018**, *99*, 2341–2352. [[CrossRef](#)]
19. Amin, A.-; Tao, W.; Doell, D.; Lingard, R.; Yin, Z.; Leu, M.C.; Qin, R. Action Recognition in Manufacturing Assembly using Multimodal Sensor Fusion. *Procedia Manuf.* **2019**, *39*, 158–167. [[CrossRef](#)]
20. Zhang, Z.; Wang, W.; Chen, Y.; Jia, Y.; Peng, G. Prediction of Human Actions in Assembly Process by a Spatial-Temporal End-to-End Learning Model. In *SAE Technical Paper 2019-01-0509*; SAE International: Warrendale, PA, USA, 2019. [[CrossRef](#)]
21. Hawkins, K.P.; Vo, N.; Bansal, S.; Bobick, A.F. Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration. In Proceedings of the 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Atlanta, GA, USA, 15–17 October 2013; pp. 499–506. [[CrossRef](#)]
22. Zhang, J.; Wang, P.; Gao, R.X. Hybrid machine learning for human action recognition and prediction in assembly. *Robot. Comput. Manuf.* **2021**, *72*, 102184. [[CrossRef](#)]
23. Dojer, N.; Gambin, A.; Mizera, A.; Wilczyński, B.; Tiuryn, J. Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinform.* **2006**, *7*, 249. [[CrossRef](#)]
24. Zou, M.; Conzen, S.D. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* **2005**, *21*, 71–79. [[CrossRef](#)]
25. Sun, J.; Sun, J. A dynamic Bayesian network model for real-time crash prediction using traffic speed conditions data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 176–186. [[CrossRef](#)]
26. Beckmann, J.; Koop, G.; Korobilis, D.; Schüssler, R.A. Exchange rate predictability and dynamic Bayesian learning. *J. Appl. Econ.* **2020**, *35*, 410–421. [[CrossRef](#)]
27. Du, Y.; Chen, F.; Xu, W.; Li, Y. Recognizing Interaction Activities using Dynamic Bayesian Network. In Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006; Volume 1, pp. 618–621. [[CrossRef](#)]
28. Frigault, M.; Wang, L.; Singhal, A.; Jajodia, S. Measuring network security using dynamic bayesian network. In Proceedings of the 4th ACM workshop on Quality of Protection, Alexandria, VA, USA, 27 October 2008; pp. 23–30. [[CrossRef](#)]
29. Chapelle, O.; Zhang, Y. A dynamic bayesian network click model for web search ranking. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 1–10. [[CrossRef](#)]
30. Gellert, A.; Zamfirescu, C.-B. Using Two-Level Context-Based Predictors for Assembly Assistance in Smart Factories. In *Intelligent Methods in Computing, Communications and Control. ICCCC 2020. Advances in Intelligent Systems and Computing*; Dzitac, I., Dzitac, S., Filip, F., Kacprzyk, J., Manolescu, M.J., Oros, H., Eds.; Springer: Cham, Switzerland, 2021; Volume 1243. [[CrossRef](#)]
31. Gellert, A.; Zamfirescu, C.-B. Assembly support systems with Markov predictors. *J. Decis. Syst.* **2020**, *29*, 63–70. [[CrossRef](#)]
32. Precup, S.-A.; Gellert, A.; Dorobantiu, A.; Zamfirescu, C.-B. Assembly Process Modeling through Long Short-Term Memory. In *Recent Challenges in Intelligent Information and Database Systems, Proceedings of the 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, 7–10 April 2021*; Hong, T.-P., Wojtkiewicz, K., Chawuthai, R., Sitek, P., Eds.; Springer: Singapore, 2021; Volume 1371, pp. 28–39. [[CrossRef](#)]
33. Ghahramani, Z. Learning Dynamic Bayesian Networks. In *Adaptive Processing of Sequences and Data Structures. NN 1997. Lecture Notes in Computer Science*; Giles, C.L., Gori, M., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1387, pp. 168–197. [[CrossRef](#)]
34. Ankan, A.; Panda, A. pgmpy: Probabilistic Graphical Models using Python. In Proceedings of the 14th Python in Science Conference, Austin, TX, USA, 6–12 July 2015; pp. 6–11. [[CrossRef](#)]
35. Ganesh, S.; Cave, V. *p*-values, *p*-values everywhere! *N. Z. Vet. J.* **2018**, *66*, 55–56. [[CrossRef](#)] [[PubMed](#)]
36. OpenVINO. Available online: <https://docs.openvino.ai/latest/index.html> (accessed on 6 December 2021).