mathematics

Article

Real-Time Assembly Support System with Hidden Markov Model and Hybrid Extensions

Arpad Gellert [1], Stefan-Alexandru Precup [1], Alexandru Matei [1], Bogdan-Constantin Pirvu [2] and Constantin-Bala Zamfirescu [1,*]

[1] Computer Science and Electrical Engineering Department, Lucian Blaga University of Sibiu, 550025 Sibiu, Romania; arpad.gellert@ulbsibiu.ro (A.G.); stefan.precup@ulbsibiu.ro (S.-A.P.); alex.matei@ulbsibiu.ro (A.M.)

[2] Industrial Engineering and Management Department, Lucian Blaga University of Sibiu, 550025 Sibiu, Romania; bogdan.pirvu@ulbsibiu.ro

* Correspondence: constantin.zamfirescu@ulbsibiu.ro

Abstract: This paper presents a context-aware adaptive assembly assistance system meant to support factory workers by embedding predictive capabilities. The research is focused on the predictor which suggests the next assembly step. Hidden Markov models are analyzed for this purpose. Several prediction methods have been previously evaluated and the prediction by partial matching, which was the most efficient, is considered in this work as a component of a hybrid model together with an optimally configured hidden Markov model. The experimental results show that the hidden Markov model is a viable choice to predict the next assembly step, whereas the hybrid predictor is even better, outperforming in some cases all the other models. Nevertheless, an assembly assistance system meant to support factory workers needs to embed multiple models to exhibit valuable predictive capabilities.

Keywords: assembly support systems; hidden Markov models; prediction by partial matching; hybrid prediction

MSC: 65C20

check for updates

Citation: Gellert, A.; Precup, S.-A.; Matei, A.; Pirvu, B.-C.; Zamfirescu, C.-B. Real-Time Assembly Support System with Hidden Markov Model and Hybrid Extensions. *Mathematics* **2022**, *10*, 2725. https://doi.org/10.3390/math10152725

Academic Editors: Alex Doboli, Daniel-Ioan Curiac, Mihai-Victor Micea and Cristina Sorina Stângaciu

Received: 4 July 2022
Accepted: 29 July 2022
Published: 2 August 2022

1. Introduction

Industry 5.0 [1,2] in Europe and Society 5.0 [3] in Japan are visions of a future super smart society in which human centricity and sustainability are key goals, going beyond efficiency and productivity, as in, e.g., Industry 4.0. Despite progress in ITC and automation technology, manual work is still an important part of European production, being the second largest category within the manufacturing sector [4,5].

Enabling workers to train and improve their skills is an important aspect within any factory. This is even more important in the context of mastering mass-customization with lot size one production, shorter production life cycles and global competition as well. Because automation is expensive, and due to flexibility and adaptivity requirements, nowadays, modern factories tend more and more to provide support for their workers through assistance systems. To execute the job correctly, these systems support the user with the right information when it is necessary, and is transmitted in a proper way (e.g., audio, video, animation, etc.), as in the Operator 4.0 concept [6]. Such a real-time assembly support system was developed and evaluated in our earlier works. The whole system was described in detail in [7]. Thus, the description of the system itself is not the goal of this paper, rather the focus being on its prediction component.

Several machine learning techniques have already been evaluated for their assembly process modeling capabilities. Thus, two-level context-based predictors were presented in [8]. The current assembly context taken from the first level's left shift register is searched in the second level's pattern history table. If it is found, the assembly state paired in the table to the current pattern is returned as the predicted next assembly state. The difference between the current state and the predicted next state gives the predicted next assembly step. The model needs a training stage, after which it can be used to predict scenarios learned before. The great disadvantage of this simple model is that it cannot keep more than one state for a certain pattern and, thus, multiple prediction choices are not possible. This drawback is eliminated by the Markov model presented in [9,10]. However, as Rabiner states in [11], Markov models in which the states correspond to observable events can be too restrictive in some applications. Therefore, the Markov model concept can be extended such that the observation becomes the state's probabilistic function. Such a doubly embedded stochastic process is called the hidden Markov model (HMM). Another extension of the Markov model is the prediction by partial matching (PPM), applied in [12], which combines multiple Markov models of different orders.

The novelty of this work consists in analyzing the applicability of HMM and its hybrid extensions for next assembly step prediction in manufacturing processes. The HMM applied with such assembly guidance purpose is an original contribution. The evaluations of the proposed method, as well as its comparisons with existing methods, will be performed on different datasets collected in experiments focused on assembling a customizable modular tablet, described in [7]. One of them consists of the data collected during an experiment involving 68 students. The other one consists of the data collected within another experiment which took place in the IFM factory and had 111 real factory workers as participants. A mixed dataset which randomly combines the assembly sequences of the students with those of the factory workers was also prepared. The HMM will be optimally configured on all these datasets and then compared with previously developed methods. As metrics, we will use the prediction rate, the coverage, and the prediction accuracy. The goal is to identify the most performant prediction method which will be integrated into the final assembly support system.

The rest of the paper is organized as follows. Section 2 reviews the latest developments in engineering assembly assistance systems and synthesizes some techniques that can be employed in this context. Section 3 presents the proposed HMM-based assembly step prediction solution and Section 4 evaluates the model and discusses the results. Finally, Section 5 concludes the paper and provides further work directions.

2. Related Work

2.1. Assembly Assistance Systems

The topic of assistance systems, from a research and application impact perspective, is summarized below, considering some of the latest articles in this field. Mark et al. [13] reviewed more than 100 papers about state-of-the-art and future directions concerning worker assistance used in manufacturing (i.e., manual work) on the shop floor. Although more industrial interest on this topic is revealed, the authors suggest that there is still a need to develop more applications for the manufacturing workers considering especially the user-interface design and biomedical aspects as well. The authors also indicate that methodology to select and evaluate the most suitable assistance systems for different worker groups is missing. Peron et al. [14] propose a decision support model to support engineers and managers to choose a cost-effective solution with assistive technologies (i.e., collaborative robots, digital instructions) in assembly operations. Throughput, worker and equipment costs, operation time and type were the variables considered by the decision support model to find out which production scenario and which configuration are the most cost effective (i.e., manual assembly, manual assembly with digital instructions, manual assembly with cobots and manual assembly with both digital instructions and cobots). The authors conclude that in order to achieve a high throughput, digital instructions and/or cobots should be considered in the assembly system, whereas for a low throughput, cobots might be considered only in high operation time situations. Reviewing more than 200 applications of assistance systems from a manufacturing process perspective, Knoke et al. [15] conclude that most of the applications are applied for assembly processes. Miqueo et al. [16] carried out a literature review on more than 200 papers looking at the effects of mass customization for assembly operations in the context of digital technologies. Whereas the key digital enabling technologies are Virtual Reality (VR) and Augmented Reality (AR), a competitive manual assembly to fulfill lot size one requirements should involve "product clustering, modularization, delayed product differentiation, mixed-model assembly, and reconfigurable assembly systems".

Assistance systems relying on AR-based technologies are the most widespread applications for assistance systems involving manual work. A real-time assistance system is presented in [17]. It can capture user motion and provide assembly assistance via AR instructions for each assembly step. Compared with paper-based instructions, the authors reveal improvements in terms of learning rate and a significant timing reduction of the first assembly procedure for the proposed assistance system. Microsoft HoloLens is a key device in another AR-based assistance system for footwear industry operators [18], deployed for an offline training of the inexperienced workers, having reduced expert interventions and lower pressure during training among its main benefits. Nevertheless, the authors reveal low usability for longer periods of using the system with Microsoft HoloLens, leading to user errors. Fu et al. present in [19] a scene-aware AR-based system, composed of a server for the image processing and a wearable Android device (i.e., AR glasses), which can recognize on the fly the assembly state and provide visual support for the operator. Deep learning was deployed with the YOLOv3 framework to recognize the manual assembly process. The system was evaluated in a 20-person experiment, clustering them into six groups executing the assembly task three times in order to find out: (a) the assembly performance for unskilled personnel using paper-based instructions versus AR-assistance, (b) the AR-device effect on skilled workers and (c) the learning time of unskilled operators with manual operations vs. the proposed system. Their system improves the assembly performance and the learning rate with respect to the paper-based approach in the case of unskilled workers. On the other hand, using AR-glasses by skilled workers reduces assembly performance. The authors also note that unskilled workers using the AR-assistance have lower accuracy compared with the skilled ones without any support. Lai et al. introduced in [20] a similar system for assisting the mechanical assembly from a CNC machine, and after comparing it with a paper-based manual they achieve more than a 30% reduction in time and errors. Generation of AR-instructions automatically from CAD models is demonstrated in [21], covering the assembly of mechanical assemblies (cables and flexible parts are not supported). A review on AR in smart manufacturing can be found in [22]. Covering more than 100 recent papers, the authors look at the devices implemented, the manufacturing process for which they serve, what function they operate and give insights to their intelligence source. The authors conclude that head-mounted displays are the most commonly used devices. Maintenance followed by assembly/disassembly are the operations where the solutions are deployed the most. Recognition and positioning are the main functions deployed. Multiple sources of intelligence account for 41% of all AR applications enclosed in the review.

VR-based assistance systems are also commonly deployed for operator assistance. Designed for the first training of workers, the virtual learning environment proposed by [23] was evaluated in a lab setting by 13 people (academia staff and students) and eight professionals. The results showed that the simulator was intuitive without generating sickness, making it suitable for initial and advanced training as well. Not surprising, more realistic interaction was one of the main immediate improvement suggestions. Gorecky et al. [24] presented a cost-effective and scalable hardware set-up with gamification features within an assistance system for the automotive industry. It is capable of automated training by loading the training scene, tools and products from existing product databases (e.g., CAD models). Recognition of human motion and prediction of the motion behavior in real-time simulation is key for human-machine collaborations, such as with robots in manual operations. For this in [25], on a shop floor, spatial region-based activity identification with real-time simulation (Unity3D) and wearable human motion captures is presented. For simple and less frequent maintenance tasks, Pratticò et al. [26] presented a VR simulator developed in Unity, using HTC Vive Pro as the head-mounted display for the users. They involved 18 people in an experiment to evaluate the training performance compared to one done just in reality. Although the training execution was faster in VR, frustration and more cognitive load were observed and improvements in the interaction with the tools and equipment would help in this regard.

Turk et al. [27] compare the impact (time, errors, ergonomics) of a smart workstation (e.g., digital instructions, pick-by-light, flexible mounting of parts and tools, etc.) versus a traditional workstation (e.g., paper-based instructions, fixed position of parts and tools, etc.) in an experiment with an unskilled worker that must perform the assembly of two products. Fewer errors, less time and better ergonomics were achieved with the smart workstation. The main drawbacks of the smart system are the lag of the software for displaying instructions and user interaction implementation (information overflow when first using the system, confusion on the current and next instruction). To achieve a worker-friendly environment and ergonomics, a multi-criteria algorithm for the monitoring and controlling of tools was further implemented by Turk et al. [28] considering worker aspects (height, gender), product complexity (precise, normal, heavy) and typology (dimension, components) and was evaluated in an experiment with 40 participants in which eight different products had to be assembled. The experiment revealed shorter assembly times and better ergonomics on the smart workstation with multi-criteria algorithm compared to the traditional one (used as benchmark). The effects of different instructions (paper-based, oral and AR-based) during the experiment on assembly operations performed by over 40 people, most of them with disabilities, is revealed by Vanneste et al. [29]. Although the instructions medium does not play a statistical role in the total assembly time, it plays a role in the number of errors made, because AR instructions generate the lowest number of errors. Looking at the stress effects, although AR instructions' score is the best, no statistical relevant differentiation between the different mediums can be determined. When looking at the perceived complexity of the task, AR-instructions deliver the best scores with statistical relevance. No statistically relevant instruction medium effects can be observed either considering the competence frustration or when looking at the perceived physical effort.

From an adaptivity perspective of the assistance system, there is strong evidence of their positive effects on operators, such as the deploying and incentive approach [30,31], adjusting the instruction length/complexity [32], adjusting the instructions in tune with the user's perceived cognitive load [33] with the help of biosensors or taking the user's experience into account for creating and presenting the instructions [34].

2.2. Prediction Techniques

HMMs, introduced in the 60s by Leonard Baum [35] and clearly described by Rabiner [11], are statistical models that are extensively used up to this day with applications in many areas of science. In the last decades, multiple types of HMMs have emerged, in examples such as first order HMM, second order HMM, superior order HMM, factorial HMM, layered HMM, hierarchical HMM or autoregressive HMM.

Usually, HMMs are used to predict or forecast different events and processes in applications such as: parsing eye tracking gaze data [36], short term load forecasting [37], reliability assessment [38] and energy consumption prediction in residential buildings [39]. HMMs are used in the automotive industry for predictive maintenance of diesel engines [40] or lane change recognition of surrounding vehicles [41]. Over the years, HMMs were also used successfully in biology and bioinformatics [42]. Applications in this field include biological sequence analysis [43–45], predict protein structure [46–48], antibiotics research [49] or to identify bacterial toxins and other characteristics in short sequence reads [50]. In cancer research, HMM models are used to predict the cell's response to drugs [51]. Training HMM models with only partially labeled data is also possible as demonstrated in [52] on both synthetic and real-world data of protein sequence. In [53], an HMM is used for hand gesture recognition. For training the model, they use gestures from the Cambridge hand gesture dataset, which has nine possible types of gestures. The first step in their approach is image pre-processing using a Sobel operator followed by a feature extraction step that selects relevant features that will be used to classify the gesture using HMM. In manufacturing, HMMs are used for life prediction of the manufacturing systems [54], fault detection [55,56] and task orchestration [57]. Authors of [58] applied a layered HMM for action recognition of a human operator assembly for human-robot collaboration to improve their cooperation. The input data is gathered using a Microsoft Kinect and two leap motion sensors and consists of the users left- and right-hand information. Another approach to human-robot collaboration for assembly is detailed in [59]. To be able to detect the operator intention but also learn continuously, an evolving HMM is used. Their experimental setup consists of a collaborative robot, different objects fitted with markers for easy pose estimation and identification, two cameras and an operation station. The data used to train the HMM model is based on the position of the building blocks, in contrast to other approaches that use the movement of the operator's hands.

Hybrid or meta-algorithmic methods usually consist of a collection of multiple prediction models that work together to give a single aggregated result. These models are not necessarily of the same type and can be trained using different methods and algorithms. One class of hybrid methods are ensemble methods that use an aggregation function to choose the final prediction such as AdaBoost, bagging or random forest (RF). AdaBoost [60] trains multiple weak classifier models on weighted input data. Weights are updated in such a way that the classifiers are focused on the misclassified examples from the previous training steps. Bagging [61] (or bootstrap aggregating) uses different subsets of the datasets to train the models that will be used to aggregate the results. RF [62] can be considered a variant of bagging where the models are trained on a trimmed-down dataset that only has a subset of the features. Aggregation strategies include [63]: arithmetic methods such as mean and median, social choice functions such as majority voting or meta-learning the aggregation function such as combiner and arbiter. In [63], an extensive study regarding aggregation strategies was made by evaluating 32 models on 46 datasets with the conclusion that there is no aggregation method among the ones evaluated that works best in any classification scenario. Depending on the stage where the aggregation is made, there can be multiple types of aggregation [64]: information aggregation, model aggregation, and hierarchical aggregation. Information aggregation can be applied when multiple sources of data are combined and then only the relevant features are selected. Model aggregation presumes multiple models are trained and then an aggregation is made on their results. Hierarchical aggregation allows information and models to be combined in a hierarchical way where the output of one model is the input for another model from a higher level in the hierarchy. Another issue in this area of research is selecting the algorithm on a meta level because usually algorithms can behave differently depending on the instance of the problem [65].

Another type of hybrid control method is to create a Markov jump system (MJS). Such systems experience large disruptive changes that makes them more efficiently described by multiple models depending on its state or operating mode. In this case, a Markov chain models the jumps between different states of the system, where each state is modeled by a different model. MJSs have various applications, usually in the field of control systems. In [66], a state-feedback controller is designed to improve stabilization of time delay MJSs in the presence of polytopic uncertainties. Authors of [67] use MJS and deep reinforcement learning to model and predict behaviors and patters in 5G networks when choosing the next target links handoffs, whereas in [68], an asynchronous fault detection mechanism is proposed for nonhomogeneous MJS. Regarding the synchronization problem of MJS, in [69], a decentralized event-triggered scheme is proposed that can decide if a sensor should communicate data at a given time to limit the used resources for a transmission.

3. Next Assembly Step Prediction through Hidden Markov Models

3.1. Assembly Support System

The assistance system (Figure 1) is a prototype developed as a flexible testbed to investigate various training scenarios involving the manual assembly work of products, from simple ones (e.g., predefined instructions, fixed order of assembly, etc.) to complex ones (e.g., instructions adapted to ongoing performance, user's state, flexible assembly recipe, etc.). In all scenarios, the modular aluminum frame with vertical adjustment of the tabletop and the large touchscreen are being used, whereas the disassembled product is placed on the tabletop. Within more complex use-cases, such as the one illustrated in Figure 1, several devices or sensors can be added to the system, such as object- and hand-tracking sensors (i.e., depth camera), posture and facial expression sensors and eye-tracking and galvanic skin response (GSR) sensors, etc.

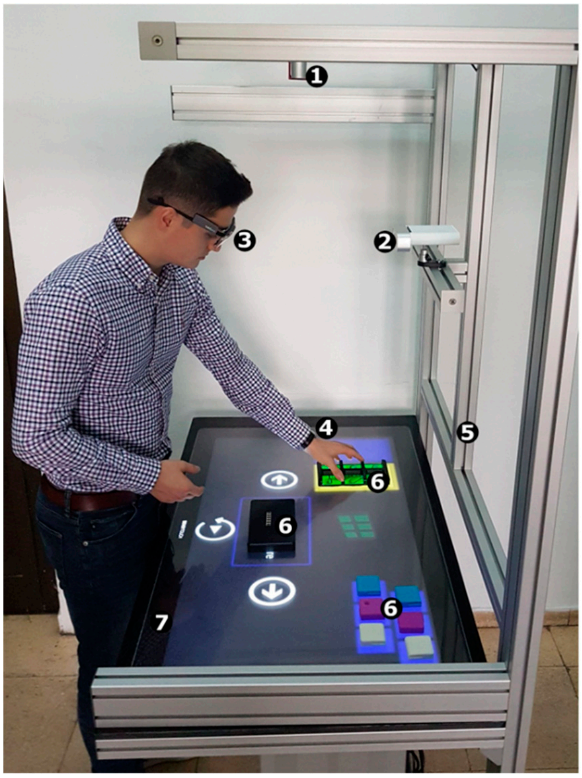

Figure 1. Manual assembly assistance system. 1: Depth camera for object detection; 2: facial expression and posture recognition sensor; 3: eye-tracking glasses; 4: GSR sensor; 5: height adjustable frame; 6: different parts of the assembled object; 7: large touchscreen.

A layout of the table with the user interaction buttons is depicted in Figure 2. The product's components are always located in the upper left and right areas of the large touchscreen before the training starts. The recommended assembly area is in the central lower part to achieve an easy reach of components and their manipulation on one side, and straightforward access to the interaction buttons on the other side. Concerning the interaction buttons, the user can control each instruction set by playing it, going to the previous or next one and even pausing it. In other use cases, the instructions can be triggered by the training application, for example when the system observes that the assembly step has been successfully completed. In the central upper side, there is the video display area, the location where videos/animations related to the assembly step are displayed. Moreover, audio instructions are played from the integrated speakers. Below the relevant components, within a given training sequence, there are visual effects (i.e., flashing), so the user can quickly identify the required components.

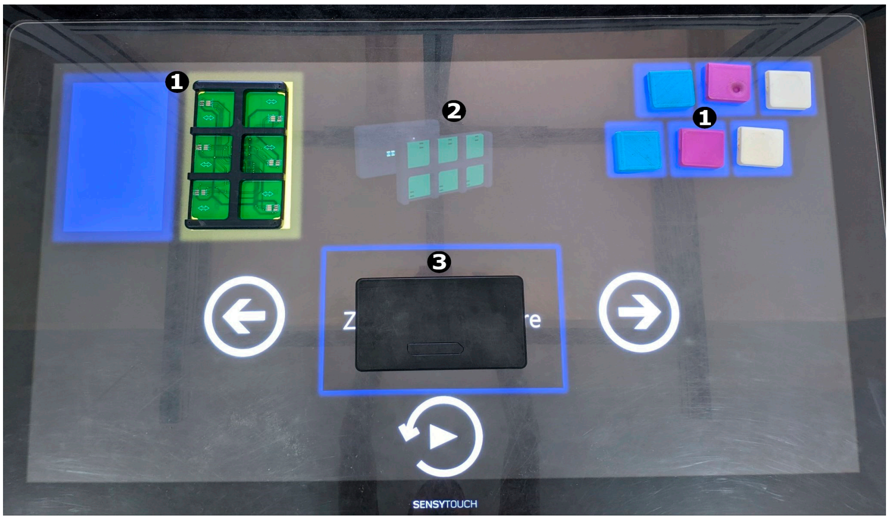

Figure 2. The graphical interface of the manual assembly assistance system. 1: parts storage area; 2: video and audio instructions; 3: workspace area.

From a software perspective, the training application utilizes the large touchscreen's hardware (see specification in Table 1) and has micro-services to support the user with adaptive instructions by changing the sequence or by modifying the content (e.g., no audio, short audio hints or detailed audio guidance; enabling closed captions for audio instructions; short animation or long detailed video instructions; time pressure enabled or not, etc.).

Table 1. Touchscreen hardware specifications.

Component	Specifications
Display	43-inch 4K touchscreen
CPU	Intel i7-7700
GPU	NVIDIA GeForce GTX 1060
RAM	16 GB
SSD	250 GB
Operating System	Windows 10

The microservices that impact the predictor and user experience are listed below:

- Tabletop height adjustment: performed manually by the user or automatically by software using the front facing camera.
- Object detection: identifies the position of each object within each image during the training.
- Depth camera streaming: provides control to the depth camera, exposing all of its capabilities such as RGB, point cloud, depth.
- Object position: establishes the 3D position of objects relying on information from the previous two services. It detects if the component was assembled correctly, and if not, it prompts the user.
- Face mimics detection: detects emotions from pictures with the user's face during training.
- Human characteristic detection: age and gender are identified from image processing of the user's picture during assembly. This service with the face mimicking service can be utilized to detect the user's state/mood.
- Predictor service: has the goal to assist the user throughout the training, providing the next best-suitable instructions depending on the user's previous and on-going performance. This is done by collecting and aggregating information from the service above with various algorithms.

All services utilize Google's Remote Procedure Call framework, have their own Service Discovery and Health Check, mechanisms that allow them to collaborate at any given time. Compared to the operator's reaction speed, the system is orders of magnitude faster (real time), minimizing the wait time for the operator between the time the current assembly task is finished and the time the next assembly task instruction is provided. More information about the concept of the prototype or other implementation details can be found in [70–72].

The product in our training use cases is a modular tablet (Figure 3), configurable in different chromatic or functional instances. It consists of one screen, one mainboard and three smaller types of modules that can be slotted in the mainboard: flashlight, speaker and battery. Due to its design, flexible assembly recipes are possible to correctly assemble the final product.

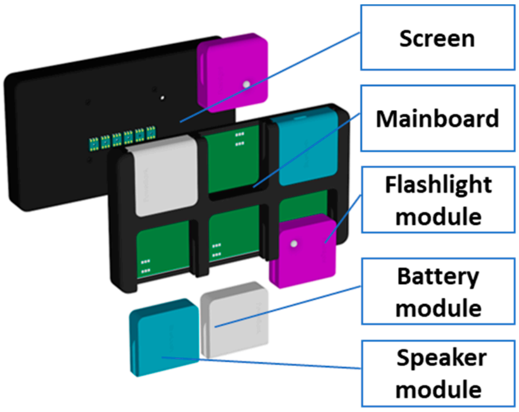

Figure 3. Example of tablet configuration as final product.

Next, this section presents the models proposed in this work to be involved in the prediction service.

3.2. The HMM-Based Prediction Algorithm

HMMs are doubly embedded stochastic processes composed of a hidden stochastic process and a set of observable stochastic processes, each state from the hidden stochastic process having associated an observable stochastic process (see Figure 4).

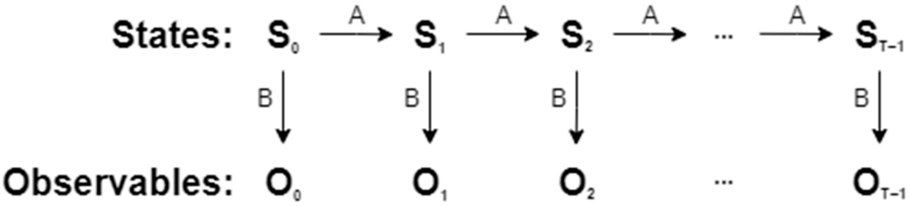

Figure 4. A generic HMM.

There are three fundamental problems that an HMM should solve. These problems were introduced by Rabiner in his work [11]. We denote the observation sequence with O and the model with $\lambda = (\pi, A, B)$, where π is the set of initial hidden state probabilities, A is the set of transition probabilities between hidden states and B is the set of observable state probabilities. The first problem that an HMM should solve is called the likelihood problem, also known as the evaluation problem. It states that, given the HMM model $\lambda = (\pi, A, B)$ and the observation sequence O, the model should be able to determine the likelihood of that observation sequence, $P(O \mid \lambda)$. The second problem, called the decoding problem, states that given the model $\lambda = (\pi, A, B)$ and observation sequence O, the model should be able to tell what was the hidden state sequence that produced the observations. In the third problem, the learning problem, given the model λ and the observation sequence O, the model should adjust its parameters $\{A, B, \pi\}$ such that $P(O \mid \lambda)$ is maximized. According to Stamp [73], the third problem is used to train the model and the first problem is used to score the likelihood for a given observation sequence.

For our application, we decided to use the HMM implementation from the Pomegranate library. Pomegranate is a Python library developed by Jacob Schreiber that, among others, offers good HMM implementation. The library offers us the possibility to train the model based on assembly sequences. The assembly steps are wrapped in an ObservableState, a wrapper class that contains information about the current assembly state and the worker's characteristics, such as the gender (0 for female, 1 for male), height (0 for short and 1 for tall), eyeglass wearer or not (1 and 0, respectively), as well as sleep quality in the preceding night (0 for bad and 1 for good). After the model is trained on the assembly sequences, it can then be used to start predicting the next piece that the worker should mount.

To predict the next piece assembled by the worker, we need the history of assemblies done on the product so far. If no piece is assembled, then the history will contain an empty assembly (since the product has just begun being assembled) with the worker's characteristics. The algorithm then proceeds to identify all the steps left to be done in the assembly sequence. After all the pieces left to be assembled have been identified, for each piece, a shadow assembly sequence containing the current assembly sequence and the piece that needs to be mounted is generated. Each shadow assembly sequence is then tested against the model to find the maximum likelihood of a shadow assembly to occur. The piece in the shadow assembly with the maximum likelihood is the one considered to be the next piece that should be assembled. A graphic representation of this method can be seen in Figure 5.

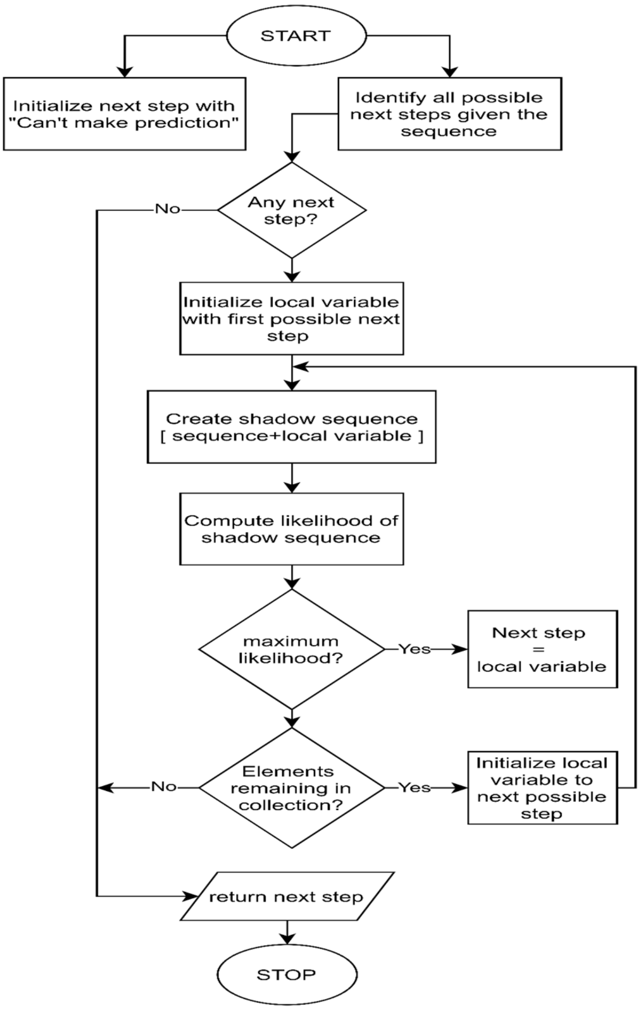
START
Initialize next step with "Can't make prediction"
Identify all possible next steps given the sequence
Any next step?
No
Initialize local variable with first possible next step
Create shadow sequence [sequence+local variable]
Compute likelihood of shadow sequence
maximum likelihood?
Yes
Next step = local variable
Elements remaining in collection?
No
Yes
Initialize local variable to next possible step
return next step
STOP

Figure 5. Prediction flowchart for HMM.

Using this approach, of determining the maximum likelihood of an assembly sequence, enables the use of this algorithm for modeling unknown (earlier unseen) assemblies. It can be seen in Section 4, that this approach is the one that will make the HMM algorithm one of the best models, compared to the other implementations.

3.3. Hybrid Prediction

Next, the question which needs an answer is: how to combine different prediction components into a hybrid predictor? First, we summarize the PPM algorithm introduced in [12] as a potential next assembly step predictor and, then, we analyze some possible hybrid schemes composed of HMM and PPM variants.

3.3.1. Prediction by Partial Matching

The PPM algorithm combines multiple Markov models of different orders. A PPM of a given order R tries to predict with the Markov model of order R and, if it cannot do that, the order of the Markov predictor is iteratively decremented until a prediction is possible. If the algorithm reaches the first order Markov model and it cannot provide a prediction, then the PPM itself is not able to predict the next state. A flowchart of the inner operations of the PPM algorithm presented in [12] can be found in Figure 6.

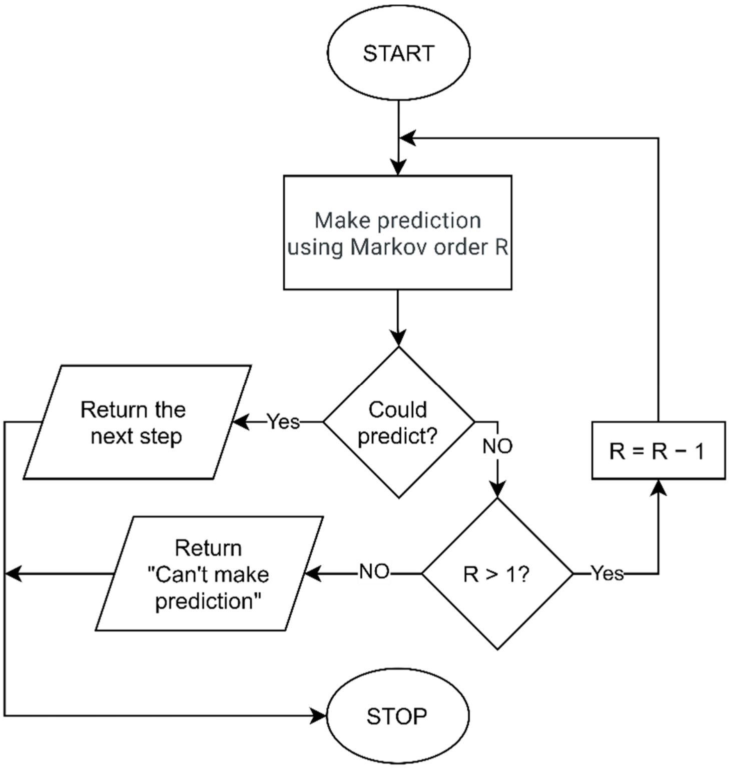

Figure 6. Prediction flowchart for PPM.

Due to the good performance of the HMM presented in Section 4, we considered the implementation of two hybrid predictors. Both these hybrid predictors internally use HMM and PPM. One hybrid predictor prioritizes the PPM to the detriment of HMM, whereas the other uses a reputation-based selection mechanism. Section 3.3.2 describes the hybrid predictor with prioritization and Section 3.3.3 the one with reputation.

3.3.2. Hybrid Predictor with Prioritization

Because we observed promising results in performance from HMM (see Section 4), we considered the implementation of a hybrid predictor. For this, we decided to pair the versatility of the HMM with the learning capabilities of the PPM algorithm introduced in [12]. For the hybridization, we considered the third order of PPM. Priority was given to the prediction of PPM whenever possible due to its higher accuracy, and in cases where PPM is not able to provide any solution, the HMM will provide one. The flow of a prediction using the hybrid predictor with prioritization is presented in Figure 7.

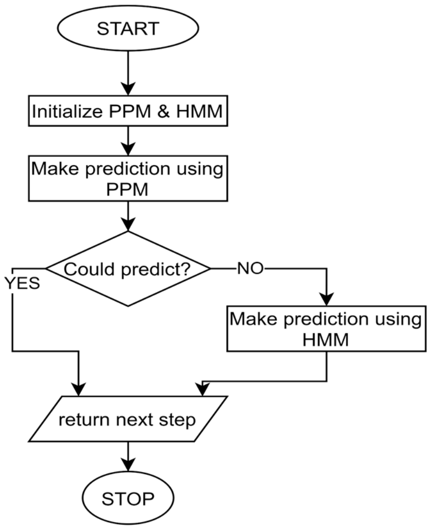

Figure 7. Hybrid predictor with prioritization.

3.3.3. Reputation-Based Hybrid Prediction

The hybrid approach, presented in the previous section, focuses on the PPM predictor, which attributes too much trust in its predictions. Compared to the previous implementation, we introduced a reputation mechanism that dictates which algorithm's prediction would be used (see Figure 8).

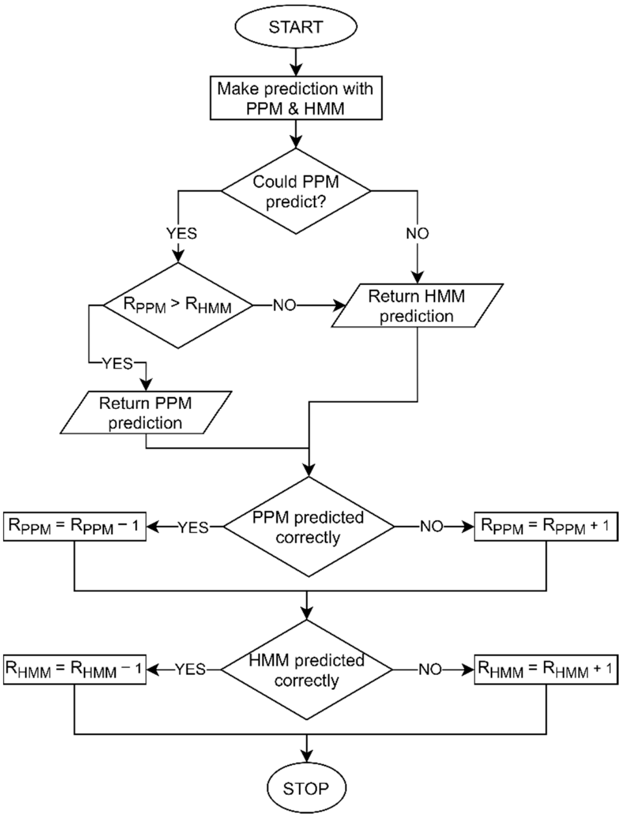

Figure 8. Hybrid predictor with reputation mechanism.

The reputation interval is in the range $[-x, x]$, where x is the maximum reputation score. The implementation of the reputation-based algorithm can be found below.

1. initialize predictors;
2. set reputation for each predictor to 0;
3. each predictor makes a prediction;
4. if only one predictor could predict, return that prediction;
5. else, select prediction based on maximum reputation;
6. receive feedback on prediction;
7. foreach predictor, if the prediction was correct, increase the reputation by 1, else decrease it by 1;
8. clamp the reputation in the reputation interval.

4. Experimental Results

Two experiments involving participants from both the academic and industrial setting were conducted, obtaining two datasets, one containing 68 assemblies from students and the other one, 111 assemblies from factory workers. Both datasets test different capabilities of the implemented algorithms. The students' dataset, due to its diversity, is useful in determining the ability of the algorithm to adapt to new scenarios, whereas the workers' dataset can highlight the learning capabilities of each algorithm. A combined analysis of the datasets will also be presented. All the participants had to assemble, without guidance, a customizable modular tablet (see Figure 3) composed of a mainboard, a screen and three types of modules (speaker, battery and flashlight).

Next, the HMM will be optimally configured on these datasets and then it will be compared with previously developed methods. We used as evaluation metrics the prediction rate, the coverage, and the prediction accuracy. The prediction rate indicates the percentage of predictions. The coverage is the percentage of correct predictions from the whole testing data. The prediction accuracy is the percentage of correct predictions from the predictions done.

The initialization phase of the HMM is heavily influenced by the randomness factor. To mitigate these influences, we decided to average the results of five randomly selected seeds. Unless specified otherwise, the results of the HMM will be presented as an average of the performance of the five seeds. There is only one parameter that can be used in the hyperparameter tuning phase, the number of the hidden states the model uses. Figure 9 shows the correlation between accuracy and the number of hidden states.

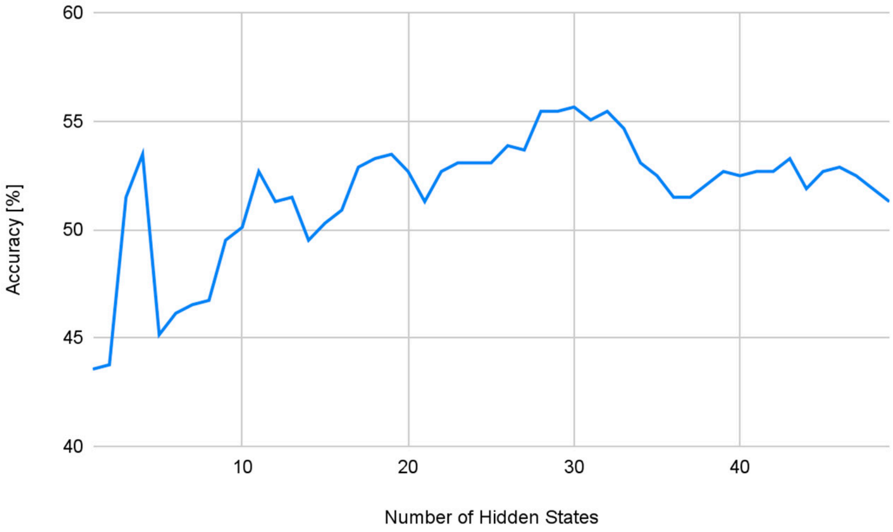

Figure 9. The correlation between accuracy and number of hidden states on the students' dataset.

There is a high percentage of correct predictions when using three or four hidden states, followed by a sharp drop in the prediction accuracy. The model continues to improve its accuracy with an increased number of hidden states, but is not able to achieve the same performance as using four hidden states. It performs better in the vicinity of 30 hidden states, with the best performance obtained by using exactly 30 hidden states, averaging an accuracy of 56.65%.

Table 2 presents the accuracy of each seed when the model is trained using 30 hidden states. As previously pointed out, the random initialization of the model provides a meaningful impact on the model's accuracy. The first two seeds, 197,706 and 20,612, have a low prediction accuracy compared to the average, with the model using the seed 197,706, performing over 5%.

Table 2. The accuracy of each seed for 30 hidden states on the students' dataset.

HMM Seed	Prediction Rate [%]	Accuracy [%]	Coverage [%]
197,706	100	51.49	51.49
20,612	100	53.47	53.47
930,364	100	56.44	56.44
938,425	100	58.42	58.42
973,051	100	58.42	58.42
Average	100	55.64	55.64

To determine if there is a significant difference using this model, we need to compare it to our previous works. In [10], we analyzed the usage of Markov predictors with different orders in the assembly context. In the same work, the Markov predictors were extended to use a padding mechanism, which significantly improved the output of the predictor. Based on these predictors, in [12], we used Markov predictors of different orders to implement a PPM mechanism. These predictors all have a problem: if a new unknown assembly scenario is presented to the model, it is unable to provide recommendations. To mitigate this issue, we extended the PPM prediction model to also investigate the neighboring states (PPMN). In [74], we used long short-term memory neural networks (LSTM) to model the assembly process, significantly improving the prediction rate compared to previous approaches. Using dynamic Bayesian networks (DBN) in [75], we increased the prediction accuracy. In [7], it was demonstrated that, through gradient boosted decision trees (GBDT), the best coverage of the assembly sequences could be obtained. In [72], an informed tree search supported by a Markov chain (A* algorithm with first order Markov model) was studied. The proposed method outperformed the decision trees (DT).

Besides the standalone HMM predictor, this work introduced four hybrid approaches: PPM supported by HMM (p_PPM_HMM), PPMN supported by HMM (p_PPMN_HMM), and their reputation-based equivalents using HMM and PPM (r_PPM_HMM) or PMMN (r_PPMN_HMM).

For the reputation-based predictor, we systematically varied the reputation interval. In Table 3, the coverage for various confidence intervals is presented on all the three datasets. The predictor was able to predict 100% of the time across all reputation intervals. Across all the datasets, the performance is quite similar. Across all the three datasets, the reputation interval that yielded the best results is [−1,1]. This will be the interval which we will use further in our comparisons.

Table 3. Coverage for different reputation intervals on all three datasets.

Reputation Interval	Students [%]	Workers [%]	Mixed [%]
$[-1, 1]$	55.45	76.45	63.32
$[-2, 2]$	55.64	76.09	62.67
$[-3, 3]$	55.25	76.69	62.45
$[-4, 4]$	55.64	76.69	62.24
$[-5, 5]$	55.45	76.69	62.31

Next, we will compare the presented methods in contrast with the best configurations of existing algorithms (the optimal order is 1 for the Markov model and 3 in the case of PPM and PPMN). First, we present the results obtained on the students' dataset.

In Figure 10, the prediction rate is compared, Figure 11 presents the coverage and Figure 12 displays the accuracy. As it can be observed, most of the predictors that rely only on previous experience, such as Markov, PPM or DBN, fail to provide any information about the probable next step, whereas predictors that infer from previous experience are able to predict 100% the time, such as HMM or its derivatives. The LSTM network is the exception, because it has a confidence interval below which it will not make a prediction.

Because DT, A*, GBDT, HMM and HMM extensions predict 100% of the time, the coverage of these models is the highest on the students' dataset. Models such as Markov chains, which have a low prediction rate also have a low coverage. If we look at the accuracy of the models, the DBN, which has the lowest prediction rate and coverage, is the one that has the highest accuracy in its predictions. The HMM models have a good coverage and accuracy, with r_PPMN_HMM having a coverage and accuracy of 58.22%.

On the workers' dataset we can observe in Figure 13 that all models are able to predict over 75% the time. This indicates that there is little variation in the assemblies of the industry workers. The coverage, presented in Figure 14, shows that the highest performing methods are still DT, HMMs and A*. DTs have a coverage of 71.18%, A* 72.35% and all HMM variants have over 75% coverage. The reputation-based hybrid using HMM and PPMN (r_PPMN_HMM) has a coverage of 78.11%.

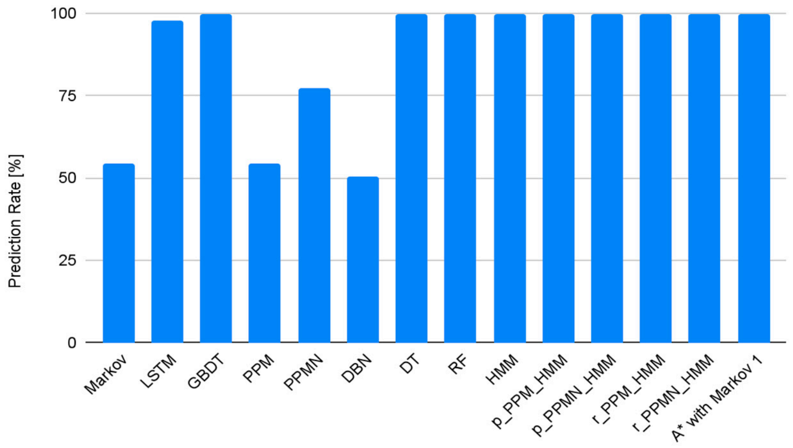

Figure 10. Prediction rate on the students' dataset.

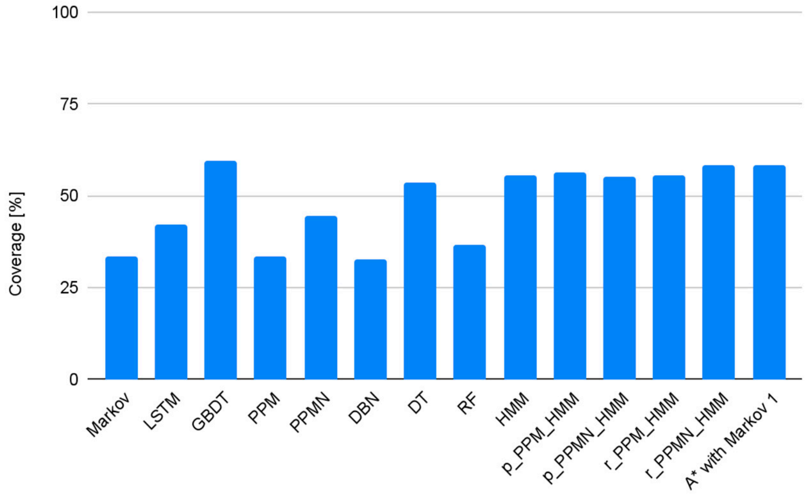

Figure 11. Coverage on the students' dataset.

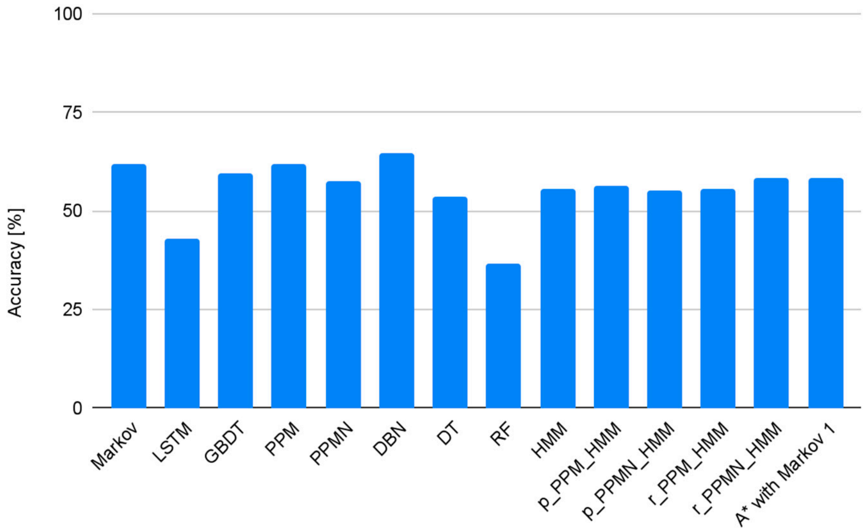

Figure 12. Accuracy on the students' dataset.

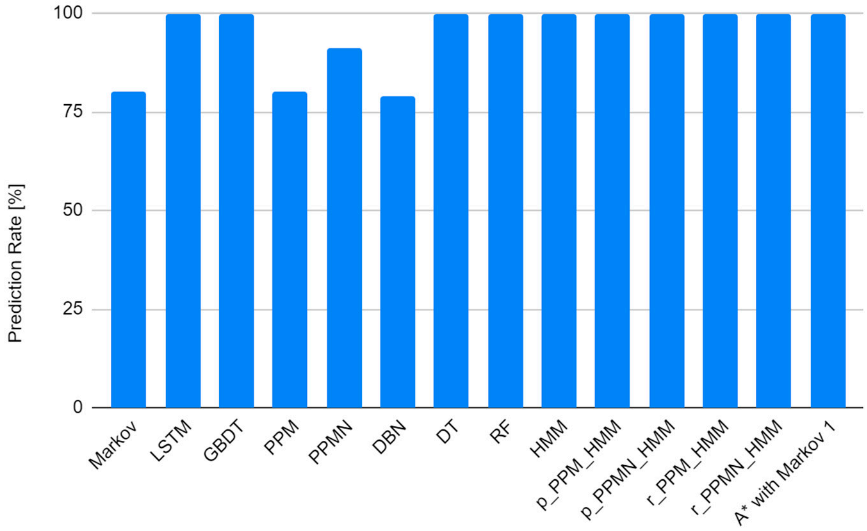

Figure 13. Prediction rate on the workers' dataset.

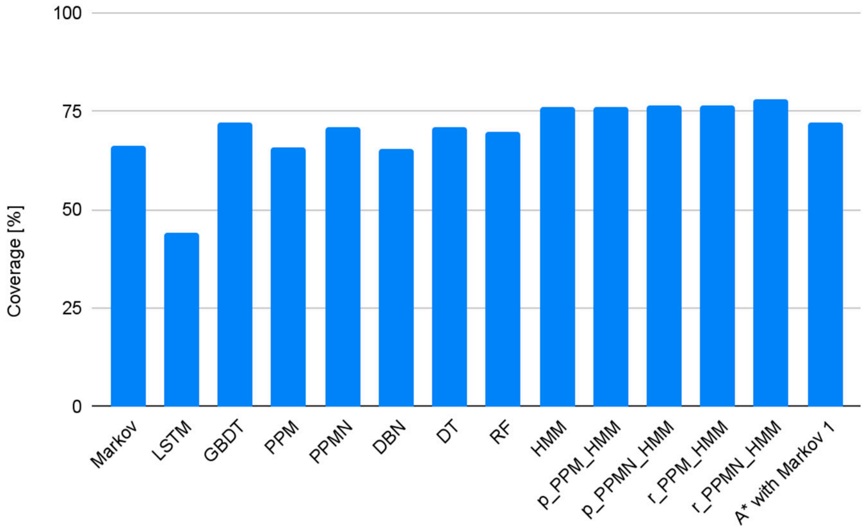

Figure 14. Coverage on the workers' dataset.

Although DT, A*, HMM and the HMM extensions have the highest coverage, it is surprising to observe in Figure 15 that the highest accuracy was obtained by the Markov, PPM or DBN models. The Markov model has an accuracy of 83.09%, PPM 82.35% and DBN 82.84%. In the cases these algorithms can predict, it is desired to use them instead of others which predict all of the time. HMM and its hybrids can predict over 76% of the cases correctly, a neglectable loss in accuracy compared to the other methods if we take into consideration that it is usable 20% more often than the others.

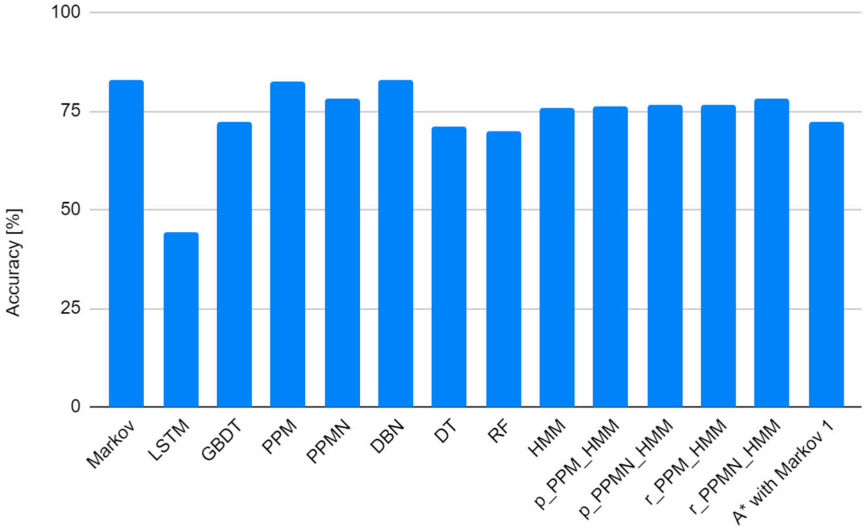

Figure 15. Accuracy on the workers' dataset.

The mixed dataset represents a combination of both students' and workers' assemblies randomly shuffled. As we can see in Figure 16, the prediction rate trend is the same across this dataset as well. Coverage wise (see Figure 17), the best performing methods are DT, HMM and its derivatives and A*. A* has a coverage of 67.63% whereas the best performing HMM-based implementation is the reputation-based hybrid that uses PPMN (r_PPMN_HMM). It has a coverage of 67.15%.

Accuracy wise, on the mixed dataset, we can observe in Figure 18 that again the Markov, PPM and DBN models have the highest accuracies, with DBN having 71.21%. A* 67.63% and r_PPMN_HMM 67.15%.

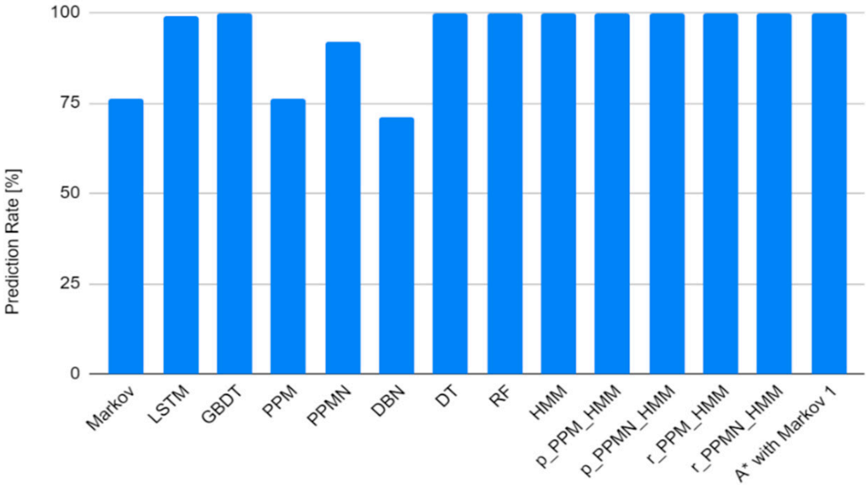

Figure 16. Prediction rate on the mixed dataset.

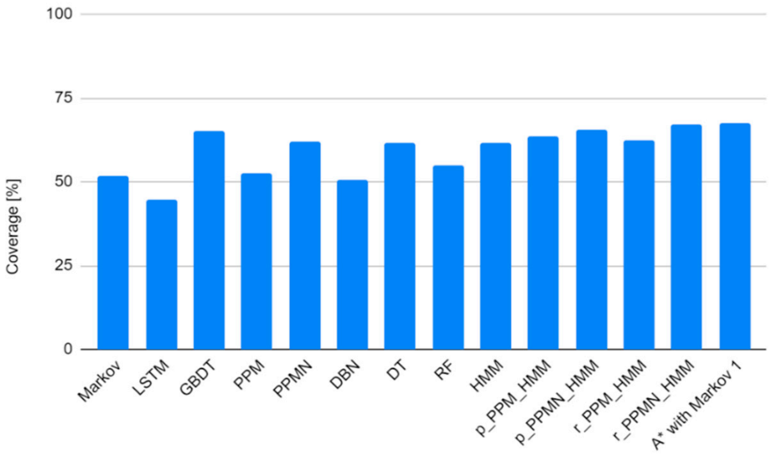

Figure 17. Coverage on the mixed dataset.

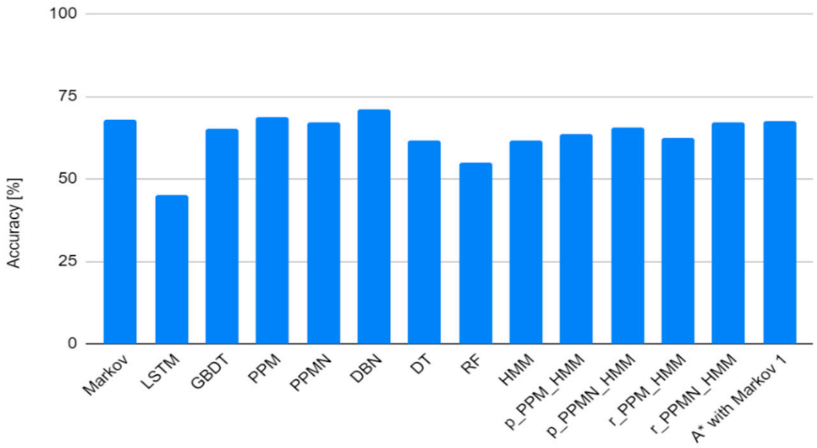

Figure 18. Accuracy on the mixed dataset.

As it can be observed, there is no algorithm that performs the best in all situations. For the students' dataset, where more distinct assemblies are presented, GBDT is the recommended method. For workers, where the data has more similarity, the HMM or one of its hybrid extensions are the best choice. For a mixed scenario, A* seems to be the best performing one. Thus, we can say that there is no approach where one algorithm fits all, and the category of workers is important when choosing the algorithm to use.

From our experiments we determined some interesting facts. The students, on average, took longer to assemble the product. Compared to the workers, there were more correct assemblies. An interesting observation is that students, perhaps due to their age, are more inclined to take alternate paths in achieving a goal, especially when assembling a product, whereas most factory workers followed the same pattern of assembly. Among the workers, there was a huge difference between those who had superior studies (bachelors, masters) and those who just completed high school. The ones with superior studies used different assembly patterns and made no mistakes when assembling the product, whereas the high school workers did not pay much attention and made several mistakes.

5. Conclusions and Further Work

In this paper, we analyzed the possibility to use HMM to predict the next manufacturing step in an industrial working environment. After the model was configured, it was compared with other existing models. Taking into account the good learning capabilities of the PPM model and the ability of the HMM to predict in all scenarios, we proposed different hybrid prediction mechanisms combining these two models. The evaluations of the proposed method, as well as its comparisons with existing methods, were performed on different datasets collected in experiments focused on assembling a customizable modular tablet. One of them consists of data from an experiment involving 68 students, whereas the other one consists of data collected in a factory from 111 workers. In addition, a mixed dataset, which randomly combines the assembly sequences of the students with those of the real factory workers, has been explored. The experimental results have shown that the HMM and its hybrid extensions are the best on the workers' dataset, with a coverage of over 75%, which is remarkable. The highest coverage was provided by the r_PPMN_HMM model at 78.11%. This model has all the qualities to be physically embedded into our final assembly support system.

There is a limitation of applying this work in industrial cases. As we previously mentioned, the students have a more creative approach in assembling the product, whereas factory workers do not pay much attention to the details and follow a predefined scheme. We believe that the accommodation of the user with such a system could potentially have a high psychological toll on them. An interesting research topic could be derived from this work: the psychological impact of assembly systems on human workers. Another technical aspect that should be considered is the nature of our modeled product. Our use case is a relatively simple one. Even though the use case is simple, the involvement of industrial partners will represent a critical point in successful adoption in the industrial context.

As a further work direction, we intend to design and evaluate other hybrid combinations as well as other selection mechanisms in hybrid prediction. For all of these, extended datasets taken from real-life assembly scenarios will be required to provide effective assistance.

Author Contributions: Conceptualization, A.G., B.-C.P. and C.-B.Z.; methodology, A.G. and S.-A.P.; software, S.-A.P.; validation, A.G. and S.-A.P.; formal analysis, A.G., S.-A.P., A.M., B.-C.P. and C.-B.Z.; investigation, A.G., S.-A.P. and C.-B.Z.; resources, B.-C.P. and C.-B.Z.; data curation, A.G. and S.-A.P.; writing—original draft preparation, A.G., S.-A.P., A.M., B.-C.P. and C.-B.Z.; writing—review and editing, A.G., S.-A.P., A.M., B.-C.P. and C.-B.Z.; visualization, A.G., S.-A.P., A.M., B.-C.P. and C.-B.Z.; supervision, A.G.; 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 the 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. Serger, S.; Tataj, D.; Morlet, A.; Isaksson, D.; Martins, F.; Mir Roca, M.; Hidalgo, C.; Huang, A.; Dixson-Declève, S.; Balland, P.; et al. *Industry 5.0, a Transformative Vision for Europe: Governing Systemic Transformations Towards a Sustainable Industry*; Publications Office of the European Union: Luxembourg, 2022. [CrossRef]
2. Maddikunta, P.K.R.; Pham, Q.-V.; Prabadevi, B.; Deepa, N.; Dev, K.; Gadekallu, T.R.; Ruby, R.; Liyanage, M. Industry 5.0: A Survey on Enabling Technologies and Potential Applications. *J. Ind. Inf. Integr.* **2022**, *26*, 100257. [CrossRef]
3. Deguchi, A.; Hirai, C.; Matsuoka, H.; Nakano, T.; Oshima, K.; Tai, M.; Tani, S. What Is Society 5.0? In *Society 5.0: A People-Centric Super-Smart Society*; Springer Singapore: Singapore, 2020; pp. 1–23. [CrossRef]
4. Chiacchio, F.; Petropoulos, G.; Pichler, D. *The Impact of Industrial Robots on EU Employment and Wages—A Local Labour Market Approach*; Bruegel: Brussels, Belgium, 2018.
5. Bisello, M.; Fernández-Macías, E.; Eggert Hansen, M. *New Tasks in Old Jobs: Drivers of Change and Implications for Job Quality*; Publications Office of the European Union: Luxembourg, 2018. [CrossRef]
6. Romero, D.; Bernus, P.; Noran, O.; Stahre, J.; Fast-Berglund, Å. The Operator 4.0: Human Cyber-Physical Systems & Adaptive Automation Towards Human-Automation Symbiosis Work Systems. In *Advances in Production Management Systems. Initiatives for a Sustainable World*; Nääs, I., Vendrametto, O., Mendes Reis, J., Gonçalves, R.F., Silva, M.T., von Cieminski, G., Kiritsis, D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 677–686. [CrossRef]

7. Sorostinean, R.; Gellert, A.; Pirvu, B.-C. Assembly Assistance System with Decision Trees and Ensemble Learning. *Sensors* **2021**, *21*, 3580. [CrossRef] [PubMed]
8. 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 International Publishing: Cham, Switzerland, 2021; pp. 167–176. [CrossRef]
9. Gellert, A.; Zamfirescu, C.-B. Assembly support systems with Markov predictors. *J. Decis. Syst.* **2020**, *29*, 63–70. [CrossRef]
10. 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]
11. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [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. Mark, B.G.; Rauch, E.; Matt, D.T. Worker Assistance Systems in Manufacturing: A Review of the State of the Art and Future Directions. *J. Manuf. Syst.* **2021**, *59*, 228–250. [CrossRef]
14. Peron, M.; Sgarbossa, F.; Strandhagen, J.O. Decision Support Model for Implementing Assistive Technologies in Assembly Activities: A Case Study. *Int. J. Prod. Res.* **2022**, *60*, 1341–1367. [CrossRef]
15. Knoke, B.; Thoben, K.-D. Training Simulators for Manufacturing Processes: Literature Review and systematisation of Applicability Factors. *Comput. Appl. Eng. Educ.* **2021**, *29*, 1191–1207. [CrossRef]
16. Miqueo, A.; Torralba, M.; Yagüe-Fabra, J.A. Lean Manual Assembly 4.0: A Systematic Review. *Appl. Sci.* **2020**, *10*, 8555. [CrossRef]
17. Pilati, F.; Faccio, M.; Gamberi, M.; Regattieri, A. Learning Manual Assembly through Real-Time Motion Capture for Operator Training with Augmented Reality. *Procedia Manuf.* **2020**, *45*, 189–195. [CrossRef]
18. Rossi, M.; Papetti, A.; Germani, M.; Marconi, M. An Augmented Reality System for Operator Training in the Footwear Sector. *Comput. Aided Des. Appl.* **2020**, *18*, 692–703. [CrossRef]
19. Fu, M.; Fang, W.; Gao, S.; Hong, J.; Chen, Y. Edge Computing-Driven Scene-Aware Intelligent Augmented Reality Assembly. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 7369–7381. [CrossRef]
20. Lai, Z.-H.; Tao, W.; Leu, M.C.; Yin, Z. Smart Augmented Reality Instructional System for Mechanical Assembly towards Worker-Centered Intelligent Manufacturing. *J. Manuf. Syst.* **2020**, *55*, 69–81. [CrossRef]
21. Neb, A.; Brandt, D.; Rauhöft, G.; Awad, R.; Scholz, J.; Bauernhansl, T. A Novel Approach to Generate Augmented Reality Assembly Assistance Automatically from CAD Models. *Procedia CIRP* **2021**, *104*, 68–73. [CrossRef]
22. Baroroh, D.K.; Chu, C.-H.; Wang, L. Systematic Literature Review on Augmented Reality in Smart Manufacturing: Collaboration between Human and Computational Intelligence. *J. Manuf. Syst.* **2021**, *61*, 696–711. [CrossRef]
23. Hirt, C.; Holzwarth, V.; Gisler, J.; Schneider, J.; Kunz, A. Virtual Learning Environment for an Industrial Assembly Task. In Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 8–11 September 2019; pp. 337–342. [CrossRef]
24. Gorecky, D.; Khamis, M.; Mura, K. Introduction and Establishment of Virtual Training in the Factory of the Future. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 182–190. [CrossRef]
25. Manns, M.; Tuli, T.B.; Schreiber, F. Identifying Human Intention during Assembly Operations Using Wearable Motion Capturing Systems Including Eye Focus. *Procedia CIRP* **2021**, *104*, 924–929. [CrossRef]
26. Pratticò, F.G.; Lamberti, F. Towards the Adoption of Virtual Reality Training Systems for the Self-Tuition of Industrial Robot Operators: A Case Study at KUKA. *Comput. Ind.* **2021**, *129*, 103446. [CrossRef]
27. Turk, M.; Resman, M.; Herakovič, N. The Impact of Smart Technologies: A Case Study on the Efficiency of the Manual Assembly Process. *Procedia CIRP* **2021**, *97*, 412–417. [CrossRef]
28. Turk, M.; Šimic, M.; Pipan, M.; Herakovič, N. Multi-Criterial Algorithm for the Efficient and Ergonomic Manual Assembly Process. *Int. J. Environ. Res. Public. Health* **2022**, *19*, 3496. [CrossRef] [PubMed]
29. Vanneste, P.; Huang, Y.; Park, J.Y.; Cornillie, F.; Decloedt, B.; Van den Noortgate, W. Cognitive Support for Assembly Operations by Means of Augmented Reality: An Exploratory Study. *Int. J. Hum. Comput. Stud.* **2020**, *143*, 102480. [CrossRef]
30. Petzoldt, C.; Keiser, D.; Beinke, T.; Freitag, M. Requirements for an Incentive-Based Assistance System for Manual Assembly. In *Dynamics in Logistics*; Freitag, M., Haasis, H.-D., Kotzab, H., Pannek, J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 541–553. [CrossRef]
31. 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*; Freitag, M., Kinra, A., Kotzab, H., Kreowski, H.-J., Thoben, K.-D., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 88–109. [CrossRef]
32. Gräßler, I.; Roesmann, D.; Pottebaum, J. Traceable Learning Effects by Use of Digital Adaptive Assistance in Production. *Procedia Manuf.* **2020**, *45*, 479–484. [CrossRef]

33. ElKomy, M.; Abdelrahman, Y.; Funk, M.; Dingler, T.; Schmidt, A.; Abdennadher, S. ABBAS: An Adaptive Bio-Sensors Based Assistive System. In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, Denver, CO, USA, 6–11 May 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 2543–2550. [CrossRef]
34. Wang, Z.; Bai, X.; Zhang, S.; Wang, Y.; Han, S.; Zhang, X.; Yan, Y.; Xiong, Z. User-Oriented AR Assembly Guideline: A New Classification Method of Assembly Instruction for User Cognition. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 41–59. [CrossRef]
35. Baum, L.E.; Petrie, T. Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **1966**, *37*, 1554–1563. [CrossRef]
36. Houpt, J.W.; Frame, M.E.; Blaha, L.M. Unsupervised parsing of gaze data with a beta-process vector auto-regressive hidden Markov model. *Behav. Res.* **2018**, *50*, 2074–2096. [CrossRef]
37. Wang, Y.; Kong, Y.; Tang, X.; Chen, X.; Xu, Y.; Chen, J.; Sun, S.; Guo, Y.; Chen, Y. Short-Term Industrial Load Forecasting Based on Ensemble Hidden Markov Model. *IEEE Access* **2020**, *8*, 160858–160870. [CrossRef]
38. Li, J.; Zhang, X.; Zhou, X.; Lu, L. Reliability assessment of wind turbine bearing based on the degradation-Hidden-Markov model. *Renew. Energy* **2019**, *132*, 1076–1087. [CrossRef]
39. Ullah, I.; Ahmad, R.; Kim, D. A Prediction Mechanism of Energy Consumption in Residential Buildings Using Hidden Markov Model. *Energies* **2018**, *11*, 358. [CrossRef]
40. Simões, A.; Viegas, J.M.; Farinha, J.T.; Fonseca, I. The state of the art of hidden markov models for predictive maintenance of diesel engines. *Qual. Reliab. Eng. Int.* **2017**, *33*, 2765–2779. [CrossRef]
41. Park, S.; Lim, W.; Sunwoo, M. Robust Lane-Change Recognition Based on An Adaptive Hidden Markov Model Using Measurement Uncertainty. *Int. J. Automot. Technol.* **2019**, *20*, 255–263. [CrossRef]
42. Eddy, S. What is a hidden Markov model? *Nat. Biotechnol.* **2004**, *22*, 1315–1316. [CrossRef] [PubMed]
43. Yoon, B.J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Curr. Genom.* **2009**, *10*, 402–415. [CrossRef] [PubMed]
44. Tamposis, I.A.; Tsirigos, K.D.; Theodoropoulou, M.C.; Kontou, P.I.; Bagos, P.G. Semi-supervised learning of Hidden Markov Models for biological sequence analysis. *Bioinformatics* **2019**, *35*, 2208–2215. [CrossRef] [PubMed]
45. Qin, B.; Xiao, T.; Ding, C.; Deng, Y.; Lv, Z.; Su, J. Genome-Wide Identification and Expression Analysis of Potential Antiviral Tripartite Motif Proteins (TRIMs) in Grass Carp (*Ctenopharyngodon idella*). *Biology* **2021**, *10*, 1252. [CrossRef]
46. Karplus, K.; Sjölander, K.; Barrett, C.; Cline, M.; Haussler, D.; Hughey, R.; Holm, L.; Sander, C. Predicting protein structure using hidden Markov models. *Proteins Struct. Funct. Bioinform.* **1997**, *29*, 134–139. [CrossRef]
47. Lasfar, M.; Bouden, H. A method of data mining using Hidden Markov Models (HMMs) for protein secondary structure prediction. *Procedia Comput. Sci.* **2018**, *127*, 42–51. [CrossRef]
48. Kirsip, H.; Abroi, A. Protein Structure-Guided Hidden Markov Models (HMMs) as A Powerful Method in the Detection of Ancestral Endogenous Viral Elements. *Viruses* **2019**, *11*, 320. [CrossRef] [PubMed]
49. Yin, X.; Jiang, X.T.; Chai, B.; Li, L.; Yang, Y.; Cole, J.R.; Tiedje, J.M.; Zhang, T. ARGs-OAP v2.0 with an expanded SARG database and Hidden Markov Models for enhancement characterization and quantification of antibiotic resistance genes in environmental metagenomes. *Bioinformatics* **2018**, *34*, 2263–2270. [CrossRef]
50. Xie, G.; Fair, J.M. Hidden Markov Model: A shortest unique representative approach to detect the protein toxins, virulence factors and antibiotic resistance genes. *BMC Res. Notes* **2021**, *14*, 122. [CrossRef]
51. Emdadi, A.; Eslahchi, C. Auto-HMM-LMF: Feature selection based method for prediction of drug response via autoencoder and hidden Markov model. *BMC Bioinform.* **2021**, *22*, 33. [CrossRef]
52. Li, J.; Lee, J.Y.; Liao, L. A new algorithm to train hidden Markov models for biological sequences with partial labels. *BMC Bioinform.* **2021**, *22*, 162. [CrossRef]
53. Sagayam, K.M.; Hemanth, D.J. ABC algorithm based optimization of 1-D hidden Markov model for hand gesture recognition applications. *Comput. Ind.* **2018**, *99*, 313–323. [CrossRef]
54. Chen, Z.; Li, Y.; Xia, T.; Pan, E. Hidden Markov model with auto-correlated observations for remaining useful life prediction and optimal maintenance policy. *Reliab. Eng. Syst. Saf.* **2019**, *184*, 123–136. [CrossRef]
55. Cheng, P.; Chen, M.; Stojanovic, V.; He, S. Asynchronous fault detection filtering for piecewise homogenous Markov jump linear systems via a dual hidden Markov model. *Mech. Syst. Signal Processing* **2021**, *151*, 107353. [CrossRef]
56. Kouadri, A.; Hajji, M.; Harkat, M.F.; Abodayeh, K.; Mansouri, M.; Nounou, H.; Nounou, M. Hidden Markov model based principal component analysis for intelligent fault diagnosis of wind energy converter systems. *Renew. Energy* **2020**, *150*, 598–606. [CrossRef]
57. Ding, K.; Lei, J.; Chan, F.T.; Hui, J.; Zhang, F.; Wang, Y. Hidden Markov model-based autonomous manufacturing task orchestration in smart shop floors. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101845. [CrossRef]
58. Berg, J.; Reckordt, T.; Richter, C.; Reinhart, G. Action recognition in assembly for human-robot-cooperation using hidden Markov models. *Procedia CIRP* **2018**, *76*, 205–210. [CrossRef]
59. Liu, T.; Lyu, E.; Wang, J.; Meng, M.Q.H. Unified Intention Inference and Learning for Human-Robot Cooperative Assembly. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 2256–2266. [CrossRef]
60. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
61. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]

62. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
63. Trevizan, B.; Chamby-Diaz, J.; Bazzan, A.L.; Recamonde-Mendoza, M. A comparative evaluation of aggregation methods for machine learning over vertically partitioned data. *Expert Syst. Appl.* **2020**, *152*, 113406. [CrossRef]
64. Feng, C.; Zhang, J. Assessment of aggregation strategies for machine-learning based short-term load forecasting. *Electr. Power Syst. Res.* **2020**, *184*, 106304. [CrossRef]
65. Tornede, A.; Gehring, L.; Tornede, T.; Wever, M.; Hüllermeier, E. Algorithm selection on a meta level. *Mach. Learn.* **2022**, 1–34. [CrossRef]
66. Alattas, K.A.; Mohammadzadeh, A.; Mobayen, S.; Abo-Dief, H.M.; Alanazi, A.K.; Vu, M.T.; Chang, A. Automatic Control for Time Delay Markov Jump Systems under Polytopic Uncertainties. *Mathematics* **2022**, *10*, 187. [CrossRef]
67. Chiputa, M.; Zhang, M.; Ali, G.G.M.N.; Chong, P.H.J.; Sabit, H.; Kumar, A.; Li, H. Enhancing Handover for 5G mmWave Mobile Networks Using Jump Markov Linear System and Deep Reinforcement Learning. *Sensors* **2022**, *22*, 746. [CrossRef]
68. Zhang, X.; Wang, H.; Stojanovic, V.; Cheng, P.; He, S.; Luan, X.; Liu, F. Asynchronous fault detection for interval type-2 fuzzy nonhomogeneous higher-level Markov jump systems with uncertain transition probabilities. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 2487–2499. [CrossRef]
69. Vadivel, R.; Ali, M.S.; Joo, Y.H. Drive-response synchronization of uncertain Markov jump generalized neural networks with interval time varying delays via decentralized event-triggered communication scheme. *J. Frankl. Inst.* **2020**, *357*, 6824–6857. [CrossRef]
70. Pîrvu, B.-C. Conceptual Overview of an Anthropocentric Training Station for Manual Operations in Production. *Balk. Reg. Conf. Eng. Bus. Educ.* **2019**, *1*, 362–368. [CrossRef]
71. Govoreanu, V.C.; Neghină, M. Speech Emotion Recognition Method Using Time-Stretching in the Preprocessing Phase and Artificial Neural Network Classifiers. In Proceedings of the 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 3–5 September 2020; pp. 69–74. [CrossRef]
72. Gellert, A.; Sorostinean, R.; Pirvu, B.-C. Robust Assembly Assistance Using Informed Tree Search with Markov Chains. *Sensors* **2022**, *22*, 495. [CrossRef]
73. Stamp, M. A Revealing Introduction to Hidden Markov Models. In *Introduction to Machine Learning with Applications in Information Security*, 1st ed.; Chapman and Hall/CRC: New York, NY, USA, 2017. [CrossRef]
74. Precup, S.-A.; Gellert, A.; Dorobantiu, A.; Zamfirescu, C.-B. Assembly Process Modeling through Long Short-Term Memory. In Proceedings of the 13th Asian Conference on Intelligent Information and Database Systems, Phuket, Thailand, 7–10 April 2021.
75. 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. [CrossRef]