Operator's Experience-Level Classification Based on Micro-Assembly Steps for Smart Factories

Fatemeh Besharati Moghaddam
Dept. of Industrial Systems Engineering and Product Design,
Ghent University
Industrial Systems Engineering (ISyE), Flanders Make
Ghent,Belgium
fatemeh.besharatimoghaddam@ugent.be

Angel J. Lopez
Dept. of Industrial Systems Engineering and Product Design,
Ghent University
Industrial Systems Engineering (ISyE), Flanders Make
Ghent,Belgium
angel.lopez@ugent.be

Stijn De Vuyst
Dept. of Industrial Systems Engineering and Product Design,
Ghent University
Industrial Systems Engineering (ISyE), Flanders Make
Ghent,Belgium
stijn.deVuyst@ugent.be

Sidharta Gautama
Dept. of Industrial Systems Engineering and Product Design,
Ghent University
Industrial Systems Engineering (ISyE), Flanders Make
Ghent,Belgium
sidharta.gautama@ugent.be

Abstract—**Tracking assembly lines in manufacturing to provide assistance is one of the essential requirement in Smart Industry. Nevertheless, these given assistance and guidelines should be offered to operators when needed. Otherwise, it can be deemed patronising in some cases, e.g., experienced operators may require less assistance than junior operators. Therefore, to provide tailored guidance and assistance in assembly lines, the operators' experience-level should be classified at different levels. In this paper, we introduce three scenarios to achieve the classification of operators expert levels in a real case study (micro-step time-series data from a factory assembly line). We implement a Convolutional Neural Network model for time-series classification, using 5 convolutional layers, max-pooling layers and 5 dense layers with dropout to avoid overfitting. We compare the results of our approach with the ground truth and also with other classifiers as K-nearest neighbours, Random Forest and Naive Bayes classifier. Results show an accuracy of 77 to 98% and 71 to 88% for two of considered scenarios.**

Index Terms—**smart assistance; time series classification; deep learning; smart factory**

I. INTRODUCTION

Industry 4.0 is transforming the next generation of manufacturing technology through intelligent processing and networking, following the goal of improving efficiency, productivity and customer satisfaction. The core element of Industry 4.0 in manufacturing is Smart industry. In smart industry, the manufacturing process move forward from traditional automation to a flexible system. Actually, the concept of smart industry is expanding from basic automation process to connecting an interface in machines to getting the data from them for decision making[20]. Today, in manufacturing line of smart industries most of the processes are planned to be done automatically by help of robots. The use of automated production lines in manufacturing remarkably reduces production and labour costs, while ensuring output consistency and quality.

But still in some parts of the line, especially in the case of assembly lines, there is a need for manual intervention by human operators.

On the other hand, depending of direct human participation, human operators may cause errors that result in unplanned downtime, low-quality or even faulty products open onto a unsatisfied customers. Adding a quality section after the manual manufacturing processing section in the factory line can be one way to reduce the human errors by evaluating the workers quality performance. Additionally, providing accurate procedures and instructions and allocating relevant and regular training and practice and personal development can be a way to reduce the human error on workplace. However, having this quality control in different steps of the assembly line can bring extra costs and human resource. Since the quality control requires human resources and human resources need to be evaluated by their performance, this just put the manufacturing line in a kind of circle.

However, the advances in technology and communications allow us to track assembly lines by means of sensors (e.g. cameras) which enable monitoring the operators allowing companies to keep the performance of efficiency of the operators high, while keeping the quality of the final product at a standard level. The functionality of sensor assistance in assembly line is to collect data by recording the detection of any changes with the help of vision detection and then by help of an interface, the evaluation of these recorded data will be possible. The micro step detection can be done in few steps. For example, in the state for *Grasp Material*, the sensor can detect the grasp, so if it is not completely done, the sensor will show the error as red light to the operator, then in assembly steps, again the sensor can detect the hand location if it is in the right place or not. This type of sensors, also connected to

projection techniques which show the exact instructions to the operator.

At the same time, operators find the sensor's process guiding in different way. This instruction set can be very valuable as a reference for inexperienced/junior operators, while this same instruction set can be annoying or obstructive to a more experienced/senior operator. So, accommodating the instructions with the experience level of the operator can be helpful in such an environment. To be certain that the right instructions will be shown to the appropriate workers, we can classify the workers by their levels of experience, so that the guidelines can be shown only to lesser experienced ones. For this classification, every workers should log in with their identity number for having access to their history by interface. As the main goal of industries is not identifying the faulty operators, but instead avoiding mistakes by operators to increase the speed and quality of assembly line, the goal of this research is to develop an efficient algorithm for the determination of the experience level of operators, without the need for personally identifiable information, by using recorded data from vision sensors. This way, the system can offer appropriate suggestion as expert level of operators so the guidelines can be shown in a smart and valid way. To achieve this, we define an approach as mentioned in section 4 with the help of Convolutional Neural Network, the evaluation will be based on historical data from a real case study as a ground truth with 3 different scenarios. Also, for showing the efficiency of CNN approach, we will compare it with 3 different classifiers.

II. LITERATURE REVIEW

During last decades, many researches are focused on machine learning algorithms for classification or clustering problems, with both supervised and unsupervised models. Sirait [21], Aamir [1], Manochandar [14] and Mishra [15] proposed clustering approaches based on different supervised and unsupervised algorithms. Recently, one of the major challenges in the field of data mining, clustering and classification, is working with time series. Time series represent a collection of values obtained from sequential measurements over time. There are various type of researches on time series related to representation, visualization and forecasting [4, 24, 11]. Esling [5] presented the state of art related to time series in data mining. They divided the literature of time series articles into three categories; representation methods (a reduced dimensionality model while keeping the main characteristics) , similarity measures (finding sub-sequence based on similar specification) and indexing techniques (having an efficient setup of data for fast retrieval).

Thus, time series classification (TSC) can be a challenging problem in data mining field these days [23, 5]. TSC consists of training a classifier on a set of inputs with its corresponding labels in order to predict at test time the correct label of a previously unseen time series [3]. Geurts [9] used a pattern model for classification, focusing on accuracy and interpretability. They presented a new tool based on a piece wise constant modelling of temporal signals by regression trees, compared to two naive feature selection techniques, to handle time series in classification problems. Also, Geler [7] continued their investigation of the k-nearest neighbour classifier on time series data and the impact of various classic distance-based vote weighting schemes by considering constrained versions of four different distance measures. Also, Zhan [25] and Li [13] used the Random Forest algorithm in TSC problems in the domain of agriculture which show high accuracy of this model compared of others.

Recently, scientists moved to deep learning techniques. Deep Convolutional Neural Networks have revolutionized the field of computer vision. It is used in machine translation, learning word embedding, speech recognition and document classification. An comprehensive state of art in TSC was written by Fawaz in 2019 [6] showing the considerable impact of deep learning, especially Residual Neural Networks (RNN) and Convolutional Neural Networks (CNN), in the field of computer vision.

On the other hand, in manufacturing research area, some of researches focuses on smart industries to improve the quality and speed of the manufacturing process based on artificial intelligence techniques. Badr [2] presented how to track the construction of complex components by employing formed task models as background knowledge together with simple vision sensors. Knoch [12] showed that they can increase the accuracy of operators by using sensor detection. They followed the detection of actions by Neural Network to gather accurate data of human behaviour such as picking and placing activities. Also,Urgo [22] proposed an approach using artificial intelligence for image processing to identify the actions of the workers and to find a way for error detection. They use a hidden Markov models in their approach. In this paper, we apply CNN in a time series classification problem. After that, we will implement K-nearest neighbours , Random Forest and Naive Bayes in the data set to compare the results.

III. METHODOLOGY

Detection of human presence in modern life is one of the interesting topics in recent years. In manufacturing field [19], the detection by means of sensors or cameras can prepare a time series data set which helpful to evaluate the error rate of operators. In this paper, we use industrial data gathered by Arkite. Arkite is a technology company that specializes in guiding operators in the manufacturing industry. They are developing a standalone device with a smart 3D vision sensor that is used to guide operators in the manufacturing industry with projection techniques to display instructions and feedback in augmented reality, directly on the work piece in the work station, so that less training is required for complex assemblies. Their engineers developed a rule-based approach that offers more or less instructions to the operators using a static number of parameters and thresholds. The approach saves all the recorded times and uses several instructions to evaluate the experience level of the workers. The Arkite approach adapted by time classified the worker's experience levels based on recorded time (the duration of time for doing

the step type assigned to each workers which will be defined in next paragraph). The current approach works based of the individual history of each worker, which means the operator must log in with their identity number and the algorithm will do the calculation according to the recorded time for the specific operator. In order to respect the privacy rules, we propose an alternative approach that uses deep learning to achieve the same result without the need for data from the individual operators.

As we mentioned before, the input of the algorithm should be related to the duration of each micro step (named micro step such as *grab* and *place* according to MTM (Methods-Time Measurement system description [8]) and each worker. Figure 1 shows the two duration of the same consecutive micro step sequence. There are two timestamps in a micro step we can measure; when the *ON* and *OFF* times are reached. As it is shown in the figure 1, a micro step divided in 2 stages as comprehend and execution. The *ON* is related to start of the execution of micro step and the *OFF* is related to end of it and the comprehend is related to the time which a worker need to understand and figure out the next steps. To better understanding, according to the setup threshold, sensor can detect the present of hand in the specific frame area of work station and record *ON* and *OFF* triggers as hand or object present and absent detection. In the top sequence everything goes as expected and the duration of each micro step can be determined as $T_{OFF} - T_{OFF-1}$. The second timeline sequence shows what can happen when a detection triggers before we expect it and before the operator is ready with his task. It shows that due to the ambiguous detection, the *OFF* time triggers are not stable. for confirm step type such as *grab*, by detection of present of hand in the frame the *ON* trigger start to be recorded and by detection of absence of hands or objects in the frame the *OFF* trigger recorded as the end of micro step. On the other hand, for some micro step such as *place* when workers need to correct something related to this micro step or redo it because of quality issue, the sensor will consider the time for next micro step and the duration time will not be accurate.

In order to reduce this noise, a step type is introduced and a naming convention is chosen for each of them. A *place* micro step has a high risk of being misinterpreted compared to a *grab* or *confirm* micro steps as we described above, which are unambiguous micro steps by nature and therefore do not trigger falsely. So, due to the ambiguous detection of some activities, they will be grouped between 2 unambiguous micro steps to form a robust step type. After detection of the *ON* and *OFF* triggers by help of vision sensors, which rely on the movement and existence of hands or objects in work station that is under supervision, the time will be recorded in an interface with the features we list below for the calculation of the duration of performing a step type.

- **Step type Label:** shows the detail of all the micro steps which combine in that step type.
- **Step type:** shows as 2 letter label such as *GE* according to factory agreement as a symbol of that step type.

- **Expert-Level:** filled automatically after the experience level calculation.
- **Assembly Step Start Time:** relates to the *ON* time.
- **Assembly Step Stop Time:** relates to the *OFF* time.
- **Break Time**: represents the time in which the operator is not available in work station.
- **Net Time:** the difference between start and stop time, which will be the duration of step type.
- **Operator ID:** the identity number of each worker.
- **Date Time:** The end time and date of the step type.

As the raw data is not suitable for the machine learning algorithms, we should perform preprocessing for the database. Data preprocessing is a vital step in any machine learning process, in which the data is transformed or encoded allow the learning algorithm to easily ingest it. In other words, the features of the data can now be easily interpreted by the algorithm. Structural and functional preprocessing of the data is performed using segmentation, normalization, balancing the data and removing noisy data. In our case, we remove the noisy data included break time which the operator stop working during a step type or the sensor can not decide if the operator is present at the work station or not. Also, We remove the maintenance, cleaning and start up processes for the work station. The next section explains the different types of normalisation and balancing techniques that we applied.

A. Model Definition

The key requirement for determining the experience level of the operators is their recorded time for each step type. It lasts from the first second of the comprehension step until the last second of the execution step. As we should consider these processing times as time series, the main challenge of our problem will be defining the experience level of operators without requiring the specific operator information using an efficient classifiers for our TSC problem.

According to [6], deep convolutional neural network can be an effective way for time series classification. A Convolutional Neural Network is a deep learning algorithm which can take in an input image, assign weights and biases to various objects in the image and be able to differentiate one from the other [16]. Convolutional layers are the major building blocks in CNN. In fact, the CNNs learn multiple features in parallel for a given input. In Neural Network, we try to build basic patterns and use those patterns for classification. We use sequence modeling and functional modeling. We implement these models with 5 convolutional layers using batch normalization for improving the performance and stability of the network, max-pooling layers to reduce the number of parameters and computations required, and also 5 dense layers with dropout to avoid over-fitting. We used *'categorical-crossentropy'* as loss function, which according to [10] is one of the most efficient loss functions for multi-class classification models. Moreover, we use the *Adam* optimizer [18] in our approach as it is one of the most important optimizer that work for almost every type of linear and non-linear problems. According to [18], the key advantage of *'Adam'* is in the choice of update step size
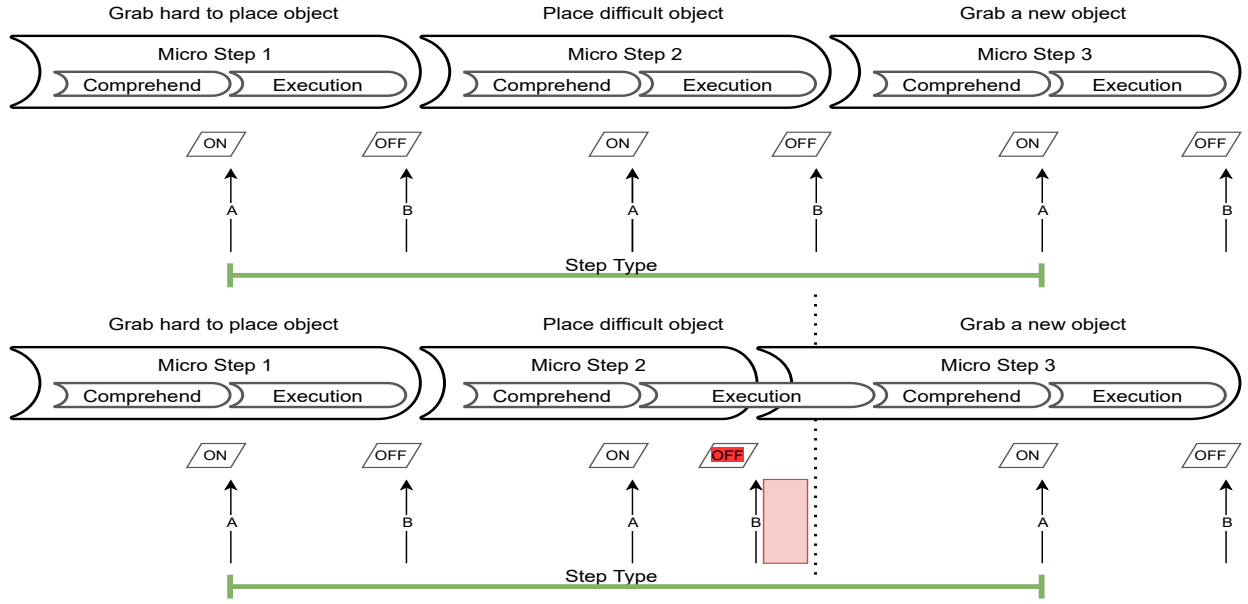
Figure 1: Step type concept according to unambiguous/ambiguous micro steps

derived from the running average of gradient moments so it is an easy approach to converge to the global minimal in fewer steps.

IV. RESULTS AND DISCUSSION

A. Experiment setup

As mentioned previously, the data set which we consider in this research is related to an assembly line of a real work place with 16 workers and 6 different step types. After preprocessing, we use the train-test-split technique in the sklearn library [17] to balance the data in the training and test sets. We further use standard scaler for normalization and K-fold cross validation with stratification to ensure that all the training and test sets have approximately the same proportions of experience levels. One of the significant hyperparameters in our approach can be time-windows (the fixed number of freshest observed entries considering as input of the classifier for each training process), which in this research we will consider 3 different amounts (3, 10 and 20) to see the classifier's result for each of them. we consider 4 different experience level's classes (class1, class2, class3 and class4) which class1 is related to least experience level and class4 is related to most expert level.

Afterwards in this approach, we consider 3 different scenarios and run them with a CNN model.The considered scenarios are defined as below:

- *scenario 1*. **considering one specific step type for one specific worker**. In this scenario we consider all the processed time recorded by the sensors for one step type and one worker to run the sequential CNN model. This scenario is the closest to the existing rule-based approach by Arkite. Each of the workers are compared only with

themselves. This method can be useful for checking the learning rate of each worker.
- *scenario 2*. **considering all activities for all workers.** Here we implement the CNN model for all 16 activities and 6 workers. In this scenario our CNN model will train just one time for the entire database, so training the model can be faster compared to the other scenarios.
- *scenario 3*. **considering one specific step type for all workers.** In the last scenario, we will check the model for all the workers but each step type separately. This scenario is the most realistic, as it will train one specific step type with all the operators, so the model can compare the experience level of all of them and according to this, decide for classification.

we implemented the aforementioned three scenarios with the CNN models. In *scenario 1*, for 16*6 times we train our CNN each time with just one step type and one worker with layers of convolutional and dense. In *scenario 2*, we just train the CNN one time with our entire data set and for *scenario 3*, for 16 times, we train our CNN each time with 1 step type and all the 6 junior/senior workers. We evaluate the hyperparameters with different amounts and after tuning them, pick the best estimated amount of them which is the nearest one to the convergence point for instance epoch=50 and train-size=0.8.

B. Experiment result

Figure 2 shows the model accuracy and loss for each of the scenarios implemented in CNN classifier. Based on the Figure 2, *scenario 1* has the highest accuracy (98% accuracy for training set and 77% accuracy for testing set). Scenario 2 has the worst accuracy (53% accuracy for training set and 52% accuracy for testing set) because of the different step types. In this scenario, CNN is training with different step type needing

rich data set (more number of workers, step type and number of recorded time for each of them), so the result will not show good performance as we can see in the figure. In *scenario 3*, our model will be trained for a specific step type for all the junior and senior workers with 88% accuracy for training set and 71% accuracy for testing set. Not only does this scenario have a good performance on the test set, in the way which CNN model receives just one step type for all the workers can be used for more reliable evaluation between operators when we have all the workers in the pool. Also as Figure 2 shows, the LOSS function presents reliable values for *scenario 1* and *scenario3*.

To illustrate the good performance of the CNN, we compare our results to two basic classification algorithms as Naive Bayes and K-nearest neighbours and also with Random Forest Classification.

In tables I and II, we can see the comparison between 4 classifiers with considering Scenario1 and Scenario3. Table I shows that *scenario 1* is better at classifying the experience levels compared to the other approaches. The same result can also been seen in *scenario 3* in Table II. The basic classifiers was not efficient for our time series classification problem.

Table I: Comparison the results of classifiers (for *scenario 1*) compared to the ground truth

Criteria	RFC	KNN	NB	CNN-S1-train	CNN-S1-test
Accuracy	61%	56%	42%	98%	77%

Table II: Comparison the results of classifiers (for *scenario 3*) compared to the ground truth

| Criteria	RFC	KNN	NB	CNN-S3-train	CNN-S3-test
Accuracy	59%	47%	40%	88%	71%

To analyze the effect of the time window, we implement these 4 classifiers for 3 scenarios with the time window set to 3, 10 and 20 respectively. The result for CNN approach is shown in Table III. The table shows that the result will be more accurate if we extend the size of the time window of our input.

Table III: Comparison of the results for different time window for each scenario.

Time window		3	10	20
Scenario1	train	60%	95%	98%
	test	56%	71%	77%
Scenario2	train	44%	50%	53%
	test	44%	48%	52%
Scenario3	train	52%	62%	88%
	test	50%	58%	71%

Additionally, as we have four different classes for experience level of operators, the accuracy of the models for each class needs to be verified. In Figure 3, we present the confusion matrix for *scenario 3*, with step type *GE* and all workers with the CNN classifier. From the figure it is clear that the

model performs acceptably for most of the classes. There is some overlap between class 3 and 4 where the CNN predicts class 4. We noticed that there is a high similarity in the recorded time patterns between classes 3 and 4, which means that operators can easily switch between these classes. This illustrates the complexity of correctly classifying these classes and is therefore a future research target.
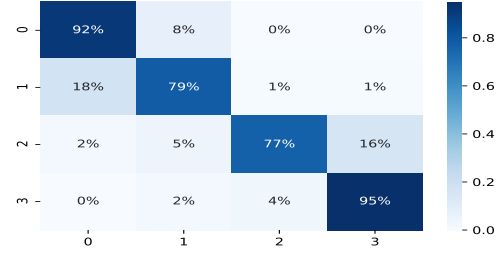
(a) confusion matrix/*scenario 3*- CNN- train
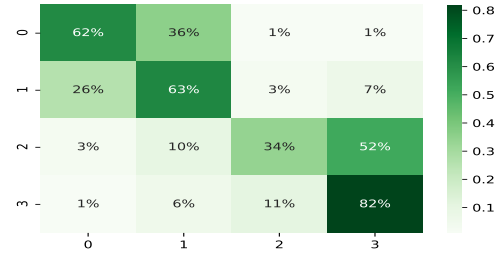
(b) confusion matrix/*scenario 3*- CNN- test

Figure 3: Confusion matrix of *scenario 3*- CNN approach for one sample step type

V. CONCLUSION

The proposed approach is suitable for Time Series Classification (TSC) on recorded times from 3D vision sensors for smart assistance based on the operator level of expert. Based on These results, the guidelines and instructions can be more reliable. We showed that a CNN model with this specific kind of input can function as an efficient classifier with an accuracy of 77 to 98% in *scenario 1* with specific step type and specific worker and 71 to 88% in *scenario 3* with specific step type and all the workers. In comparison, basic classifiers as Naive Bayes and K-nearest Neighbour and also Random Forest classifier do not present an efficient performance in the operator's experience level classification problem. Finally, we analyzed the effect of the size of the time window on the accuracy of each of the classifiers, and found that a larger entry observation as time window improves the accuracy.

In this research all of the models depend on raw data, which comes directly from the vision-sensor interface, so the future work can rely on moving to explainability to manage our data to be smarter so the improvement can happen in our TSC problem. Also, we can go deeper in our ground truth to find a pattern for distinguishing classes 3 and 4 and increase the
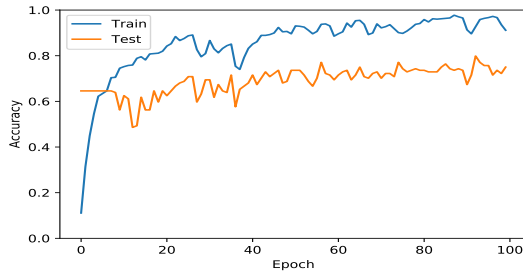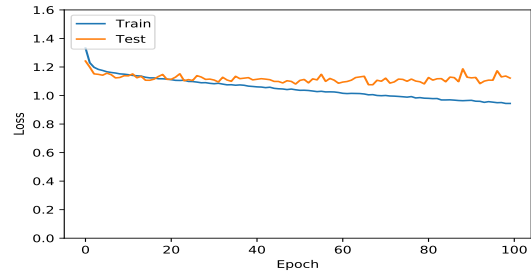
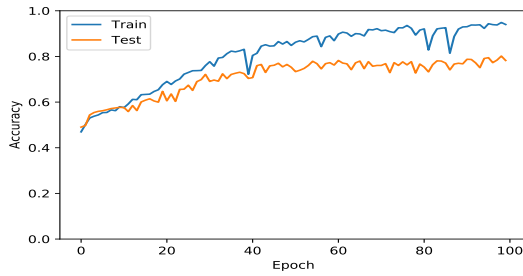(a) model accuracy for *scenario 1*
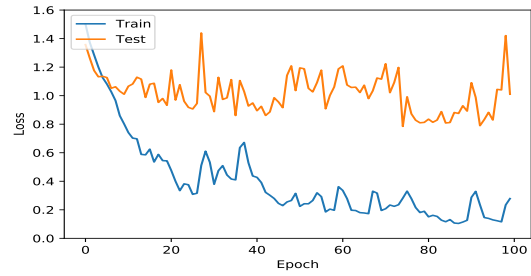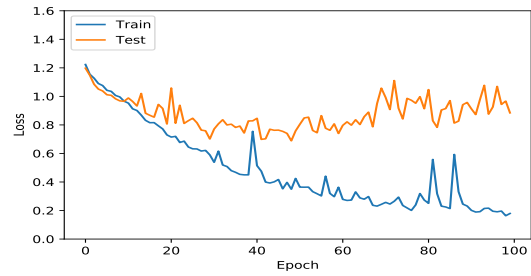
(b) model LOSS for *scenario 1*
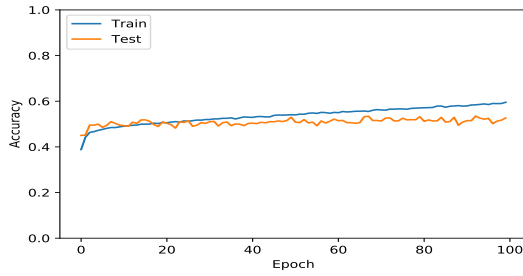
(c) model accuracy for *scenario 2*

(d) model LOSS for *scenario 2*

(e) model accuracy for *scenario 3*

(f) model LOSS for *scenario 3*

Figure 2: Comparison between *scenario 1*, *scenario 2* and *scenario 3* in CNN model for time-window=20 according to Accuracy and LOSS

accuracy of our models for the classification of these two classes.

REFERENCES

[1] M. Aamir and S. M. A. Zaidi. Clustering based semi-supervised machine learning for ddos attack classification. *Journal of King Saud University-Computer and Information Sciences*, 2019.

[2] S. Bader and M. Aehnelt. Tracking assembly processes and providing assistance in smart factories. In *ICAART (1)*, pages 161–168, 2014.

[3] B. Bai, G. Li, S. Wang, Z. Wu, and Y. Wen. Time series classification based on multi-feature dictionary representation and ensemble learning. *Expert Systems with Applications*, page 114162, 2020.

[4] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and dis-tance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[5] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[6] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[7] Z. Geler, V. Kurbalija, M. Ivanović, and M. Radovanović. Weighted knn and constrained elastic distances for time-series classification. *Expert Systems with Applications*, 162:113829, 2020.

[8] A. Genaidy, A. Agrawal, and A. Mital. Computer-ized predetermined motion-time systems in manufactur-ing industries. *Computers & Industrial Engineering*, 18(4):571–584, 1990.

[9] P. Geurts. Pattern extraction for time series classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer,

2001.

[10] I. Goodfelow, Y. Bengio, and A. Courville. Deep learning (adaptive computation and machine learning series), 2016.

[11] G. Jiang, W. Wang, and W. Zhang. A novel distance measure for time series: Maximum shifting correlation distance. *Pattern Recognition Letters*, 117:58–65, 2019.

[12] S. Knoch, N. Herbig, S. Ponpathirkoottam, F. Kosmalla, P. Staudt, P. Fettke, and P. Loos. Enhancing process data in manual assembly workflows. In *International Conference on Business Process Management*, pages 269–280. Springer, 2018.

[13] H. Li, C. Zhang, S. Zhang, and P. M. Atkinson. Crop classification from full-year fully-polarimetric l-band uavsar time-series using the random forest algorithm. *International Journal of Applied Earth Observation and Geoinformation*, 87:102032, 2020.

[14] S. Manochandar, M. Punniyamoorthy, and R. K. Jeyachitra. Development of new seed with modified validity measures for k-means clustering. *Computers & Industrial Engineering*, 141:106290, 2020.

[15] N. K. Mishra and P. K. Singh. Fs-mlc: Feature selection for multi-label classification using clustering in feature space. *Information Processing & Management*, 57(4):102240, 2020.

[16] M. A. Nielsen. *Neural networks and deep learning*, volume 2018. Determination press San Francisco, CA, 2015.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] S. Raschka and V. Mirjalili. *Python machine learning*. Packt Publishing Ltd, 2017.

[19] R. Refflinghaus and C. Kern. On the track of human errors-procedure and results of an innovative assembly planning method. *Procedia Manufacturing*, 21:157–164, 2018.

[20] X.-F. Shao, W. Liu, Y. Li, H. R. Chaudhry, and X.-G. Yue. Multistage implementation framework for smart supply chain management under industry 4.0. *Technological Forecasting and Social Change*, 162:120354, 2020.

[21] K. Sirait, E. B. Nababan, et al. K-means algorithm performance analysis with determining the value of starting centroid with random and kd-tree method. In *Journal of Physics: Conference Series*, volume 930, page 012016. IOP Publishing, 2017.

[22] M. Urgo, M. Tarabini, and T. Tolio. A human modelling and monitoring approach to support the execution of manufacturing operations. *CIRP Annals*, 68(1):5–8, 2019.

[23] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[24] C. Yiakopoulos, K. Gryllias, M. Chioua, M. Hollender, and I. Antoniadis. An on-line sax and hmm-based anomaly detection and visualization tool for early disturbance discovery in a dynamic industrial process. *Journal of Process Control*, 44:134–159, 2016.

[25] Y. Zhan, S. Muhammad, P. Hao, and Z. Niu. The effect of evi time series density on crop classification accuracy. *Optik*, 157:1065–1072, 2018.