Machine Learning Based Design Patterns Prediction

Dr. Pooja Krishnamurthy Revankar[1]
[1] VLSI & AIML Trainer, BJV Tech Hub, Ranebennur, Karnataka, India;
Email: pujarevankar09@gmail.com

Dr. Srinivasarao Udara[2]
[2]Associate Professor, Department of ECE,
S. T. J. Institute of Technology, Ranebennur, Karnataka, India;
Email: drsrudara@gmail.com

Dr. Jayprabha Vishal Terdale[3]
[3]Assistant Professor, Dept of EXTC,
A. C. Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India;
Email: jterdale@gmail.com

Prof. Gnaneshwara M P[3]
[3]Assistant Professor, Dept of ECE,
S. T. J. Institute of Technology, Ranebennur, Karnataka, India;
Email: gnaneshwara2007@gmail.com

Article History: Received : 08-08-2021 Revised: 06-09-2022 Accepted: 18-11-2022

Abstract

Although machine learning methods for industrial maintenance systems have been extensively studied in recent years, their practical adoption remains slow. One key challenge is the lack of comparative analyses of machine learning systems. To address this, we conducted a systematic literature review (2012–2021) of 104 monitoring and prediction systems. Additionally, we identified five design patterns—high-level construction frameworks—through k-means cluster analysis. Our findings indicate that while monitoring and prediction systems vary in their operational choices, they typically employ similar learning strategies (e.g., supervised learning) and tasks (e.g., classification, regression). This study aims to provide researchers and practitioners with insights into common characteristics, contextual applications, and emerging trends.

Keywords: **M L, Classification, regression**

DOI Number: 10.48047/NQ.2022.20.13.NQ88537 Neuroquantology 2022; 20(13): 4446-4457

1. Introduction

The digitalization of manufacturing machines, such as cyber-physical systems, has led to an exponential increase in data generation (Tang et al., 2014). This data holds significant value for machine learning systems (MLS), enabling applications like monitoring and prediction (Monostori et al., 2016). Given that machine maintenance accounts for a substantial portion of manufacturing costs (Mobley, 2002), the demand for monitoring and prediction systems—such as condition monitoring (CM) and predictive maintenance (PM)—has grown in recent years (Motaghare et al., 2018; Serin et al., 2020). However, their practical implementation in manufacturing remains slow, partly due to a lack of understanding of relevant technological layers (Schuh et al., 2020).

Despite extensive research on maintenance systems, there is no comprehensive systematization of MLS that provides a standardized tool for extracting design patterns representing their technological structure. We define a **design pattern** as a systematized representation of the most frequently occurring characteristics of a specific software system—a high-level construction manual that serves as a decision aid for system design and development, incorporating key MLS-related properties. Since similar problems often lead to similar solutions, these patterns offer high transferability, bridging the gap between theoretical MLS research and practical implementation. Additionally, design patterns serve as a valuable foundation for the engineering of MLS software, benefiting both researchers and practitioners (Schuh et al., 2020).

2. Literature Survey

Machine learning (ML) has become an essential tool for predicting design patterns in software and system development. By analyzing large datasets of existing software architectures, ML models can identify recurring structural patterns that enhance system efficiency, maintainability, and scalability. Supervised learning techniques, such as decision trees, support vector machines (SVM), and deep learning models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are commonly used to classify and predict design patterns. Feature extraction methods, including natural language processing (NLP) for code analysis and graph-based representations of software structures, further improve prediction accuracy. The integration of ML in design pattern prediction enables automated code refactoring, assists developers in selecting optimal patterns, and reduces software development time.

Despite these advancements, challenges remain in ML-based design pattern prediction. Variability in coding styles, inconsistent documentation, and the need for large, well-annotated datasets pose significant obstacles. Additionally, ML models must generalize well across different programming paradigms and frameworks to be truly effective. Transfer learning and reinforcement learning techniques are being explored to address these issues, allowing models to adapt to new design patterns with minimal retraining. Future research focuses on improving interpretability, ensuring robustness, and integrating ML-driven pattern prediction into real-time software development environments, making intelligent design pattern selection a seamless part of the coding process.

eISSN: 1303-5150

To aware of related work in our subject areas. However, none of these publications provides a com- parable overview of the technological structure from a technology management perspective. In contrast, our study is based on a successfully evaluated frame- work that has been developed by practitioners and re- searchers to identify design patterns. In the following we present a selection of the closest related work. Monitoring Systems.

[1] Serin et al. (2020) employed study to summarize CM systems using deep learn- ing and described opportunities of CM in Industry 4.0 environments.

[2] Mohanraj et al. (2020) presented a literature review focusing on monitoring methods for tool CM in the milling process which have been de- scribed in manufacturing-related literature.

[3] Soother and Daudpoto (2019) as well as Choudhary et al.(2019) reviewed CM techniques in an engineering context (induction motors).

[4] Choudhary et al. (2019) focused more on the fault diagnosis pro- cess and diagnostic schemes. Prediction Systems.

[5] Zonta et al. (2020) presented a literature review of PM applications in Industry 4.0 environments focusing on existing methods, stan- dards, and applications.

[6] Carvalho et al. (2019) published a literature review of ML methods focusing on their performance, challenges, and opportunities in the context of PM.

[7] Theissler et al. (2021) surveyed PM approaches in the automotive industry from an application and ML perspective.

[8] C¸ ınar et al. (2020) reviewed smart manufacturing PM applications by focusing on ML-related criteria, such as data acquisition and data classification.

3. Proposed Methodology

The proposed system for Machine Learning-Based Design Patterns Prediction aims to enhance the efficiency and accuracy of detecting and recommending design patterns in software development. It utilizes machine learning algorithms trained on large datasets of annotated software projects to predict design patterns automatically.

By analyzing software metrics, textual requirements, and structural features, the system identifies suitable design patterns and their roles in software architecture. Advanced techniques such as natural language processing (NLP) for text-based input and feature extraction for software metrics are incorporated to improve prediction reliability.

This system reduces manual effort, streamlines decision-making in design, and ensures consistency in applying patterns across diverse software domains. It is particularly beneficial for large-scale projects where complex interactions between components make manual pattern detection challenging.

eISSN: 1303-5150

4. Implementation

To implement this project we have designed following modules

Tools and Libraries: The project was implemented using Python with machine learning libraries like scikit-learn for traditional models and TensorFlow for neural networks.

Data Preprocessing: Data normalization and scaling were applied to ensure uniform input for model training. Ensemble and tree-based models were chosen for their interpretability in software metric applications.

Training Setup: Models were trained and validated using balanced splits of the dataset. Hyperparameter tuning was conducted to optimize model performance.

Performance Metrics:

Accuracy: The best-performing models achieved 83% accuracy.

Precision and Recall: Consistent values across MVP and MVVM classifications, showing stable model performance.

F1-Score: The highest F1-scores were achieved by models like Random Forest and SVM, indicating balanced sensitivity and specificity.

Model Interpretability:

Tree-based models (Random Forest, CatBoost) highlighted the importance of metrics like the number of default methods and return statements in distinguishing MVP and MVVM.

Strengths: The approach demonstrates high accuracy and interpretability, with models reliably classifying MVP and MVVM patterns based on relevant code metrics.

The machine learning industry is young and is continuously changing. Rapid hit and trial experiments are being carried on. The process is transforming, and hence there are high chances of error which makes the learning complex. It includes analyzing the data, removing data bias, training data, applying complex mathematical calculations, and a lot more. Hence it is a really complicated process which is another big challenge for Machine learning professionals.

Training Data

The most important task you need to do in the machine learning process is to train the data to achieve an accurate output. Less amount training data will produce inaccurate or too biased predictions. Let us understand this with the help of an example. Consider a machine learning algorithm similar to training a child. One day you decided to explain to a child how to distinguish between an apple and a watermelon. You will take an apple and a watermelon and show him the difference between both based on their color, shape, and taste. In this way, soon, he will attain perfection in differentiating between the two. But on the other hand, a machine-learning algorithm needs a lot of data to distinguish. For complex problems, it may even require millions of data to be trained. Therefore we need to ensure that Machine learning algorithms are trained with sufficient amounts of data.

eISSN: 1303-5150

This is one of the common issues faced by machine learning professionals. The machine learning models are highly efficient in providing accurate results, but it takes a tremendous amount of time. Slow programs, data overload, and excessive requirements usually take a lot of time to provide accurate results. Further, it requires constant monitoring and maintenance to deliver the best output.

Imperfections in the Algorithm When Data Grows

So you have found quality data, trained it amazingly, and the predictions are really concise and accurate. Yay, you have learned how to create a machine learning algorithm!! But wait, there is a twist; the model may become useless in the future as data grows. The best model of the present may become inaccurate in the coming Future and require further rearrangement. So you need regular monitoring and maintenance to keep the algorithm working. This is one of the most exhausting issues faced by machine learning professionals.

5. Results and Discussions

Successful software development fully dependent on Design Patterns as this reduces development work by reusing already developed software's functions. Incorrect design patterns often leads to failure and inexperience often fall prey for incorrect design pattern selection.

To overcome from this issues we are employing Machine Learning and WEB modules functions which will read source code as input from the user through web interface and then ML algorithms will rank source code to find suitable design pattern and then display predicted design pattern as output.

This ML models can be applied on UI/NON-UI based design patterns selection and for accurate selection we have evaluated performance of multiple ML algorithms like SVM, Random Forest and Decision Tree and each algorithm performance is evaluated in terms of accuracy, precision, recall and FCSORE.

To train above algorithms we have utilized Design Patterns prediction dataset downloaded from GITHUB URL and in below screen showing dataset details

eISSN: 1303-5150

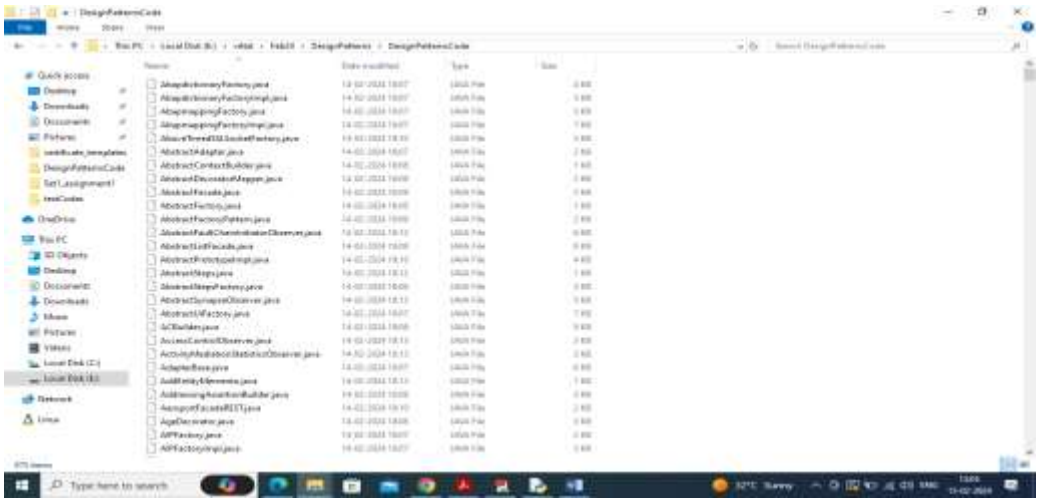

Figure 1: different design patterns

In above screen we have java code from 13 different designs patterns and all those patterns names we can see in below file

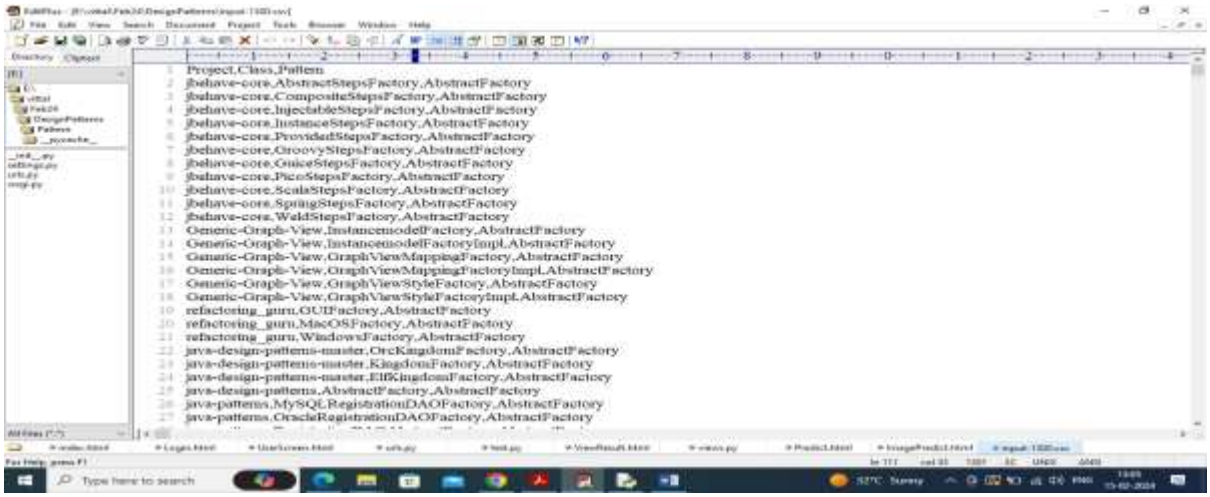

Figure 2: classification patterns

In above file first row represents column names like Project Name, source code class name and the pattern that class is following and remaining rows contains dataset values.

So by using above java source code we will train algorithms to predict design patterns. Each pattern will be selected by employing ontology based ranking calculations which will calculate rank between dataset source code and user uploaded source code and based in highest ranking Design Pattern will be selected.

By using this project developers can upload existing or current source code files and then application will predict Design Patterns and by using this prediction Developer can what type of code follow what patterns so for his next project he will choose correct patterns.

We have implemented this project as REST based web services which consists of following modules

User Login: user can login to system using username and password as ‘admin and admin’.

eISSN: 1303-5150

Load Design Patterns Code: after login user will run this module to upload dataset to application

Code to Numeric Vector: all codes will be converted to numeric vector which will replace each word occurrence with its average frequency.

Train ML Algorithms: processed numeric vector will be split into train and test with a ratio of 80:20. 80% dataset will be input to training algorithms to train a model and this model will be applied on 20% test data to calculate accuracy

Predict Design Patterns: user will upload test source code files and then ML algorithms will rank test file to predict accurate design patterns.

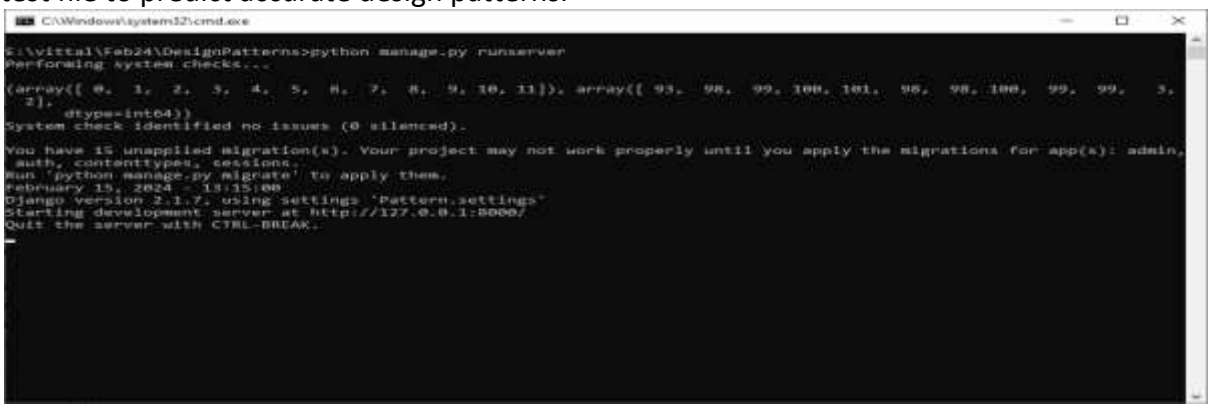

Figure 3: URL

In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below page

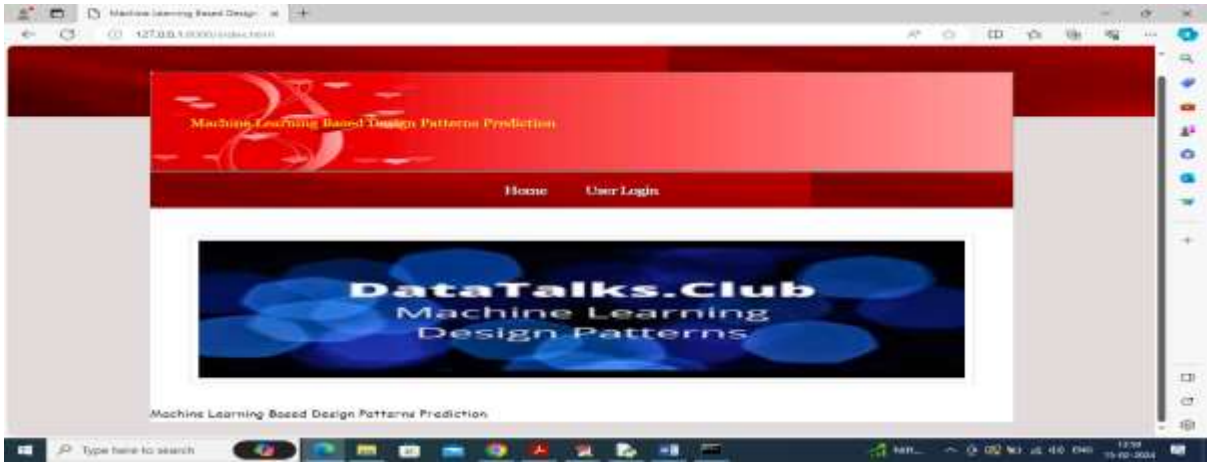

Figure 4: user login

In above screen click on 'User Login' link to get below login page

eISSN: 1303-5150

Figure 5: using username and password

In above screen user is login by using username and password as 'admin and admin' and then click on 'Login' button to get below page

Figure 6: 'Load Design Pattern Code'

In above screen click on 'Load Design Pattern Code' link to load dataset and get below output

Figure 7: Code to Numeric Vector

eISSN: 1303-5150

In above screen dataset loaded and now click on 'Code to Numeric Vector' link to convert dataset into numeric vector and get below output

Figure 8: 'Train ML Algorithms'

In above screen entire dataset converted to numeric vector and then click on 'Train ML Algorithms' link to train ML and get below output

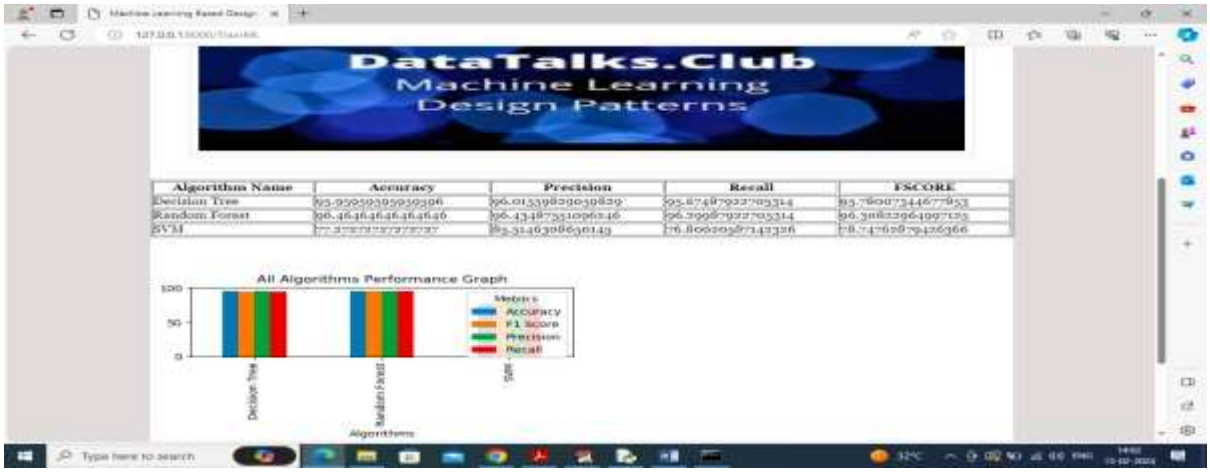

Figure 9: Predicted Design patterns

In above screen can see each algorithm performance in tabular and graph format and in all algorithms Random Forest got high accuracy and in graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and now click on 'Predict Design Patterns' link to get below page

eISSN: 1303-5150

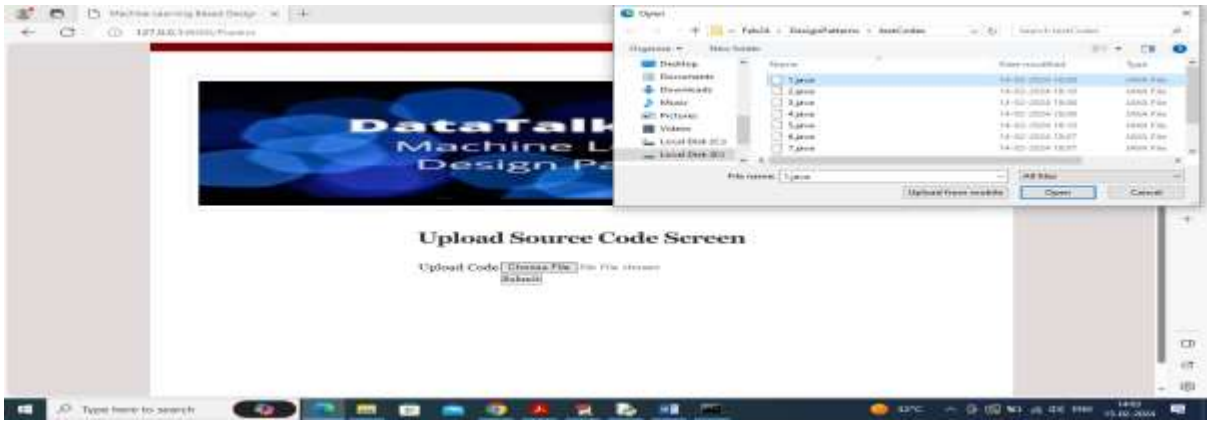

Figure 10: UI/non-UI format

In above screen select and uploading any java source code in UI/non-UI format and then click on 'Submit' button to predict names of design pattern

Figure 11: Design pattern predicted

In above screen in blue colour text can see Design pattern predicted from uploaded source code as 'Builder' and similarly you can upload and test any other source code. Below is another example

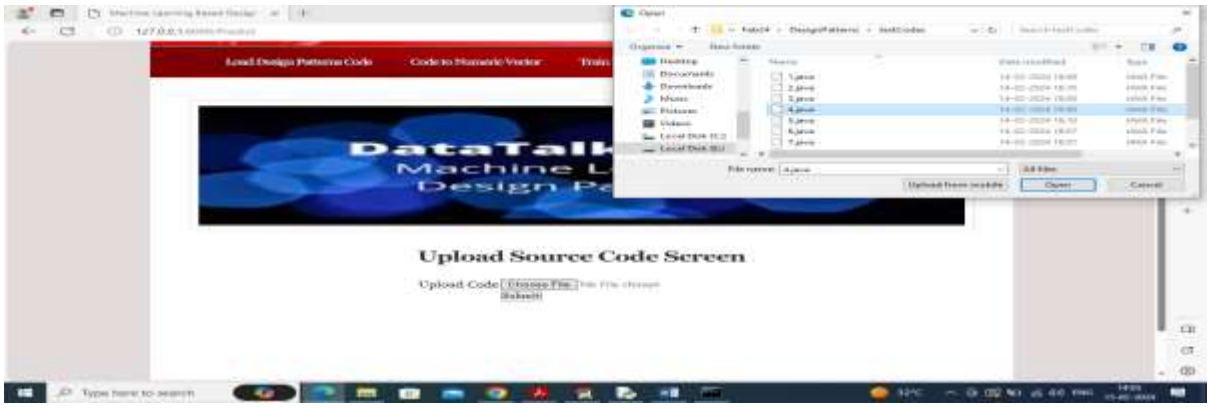

Figure 12: Uploading another code and below is the output

eISSN: 1303-5150

Figure 13: pattern detected as “Façade”

Figure 14: patterns predicted as ‘Factory Method’.

Conclusion

The presented a comprehensive overview of monitoring and prediction MLS in the manufacturing industry, i.e., performing a classification of 104 papers and extracting 5 design patterns. The conclude that research on monitoring and pre- diction MLS is quite diverse. This trend is highlighted by the growing capabilities of MLS, their operations, and tasks in recent year, especially in the field of NN. We emphasize the dependence of MLS operations and tasks on use cases, the resulting systems’ variability, as well as the high impact of data sets on the comparability of MLS. We strongly recommend continuing research, e.g., analyzing further MLS working areas and manufacturing-related diagnosis applications.

eISSN: 1303-5150

References

[1] Alegeh, N., Shagluf, A., Longstaff, A. P., and Fletcher, S. (2019). Accuracy in detecting failure in ballscrew assessment towards machine tool servitization. In- ternational Journal of Mechanical Engineering and Robotics Research, 8(5):667–673.
[2] Carvalho, T. P., Soares, F., Vita, R., Francisco, R. d. P., Basto, J., and Alcal´a, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering, 137:106024.
[3] Choudhary, A. K., Goyal, D., Shimi, S. L., and Akula, A. (2019). Condition monitoring and fault diagnosis of induction motors: A review. Archives of Computa- tional Methods in Engineering, 26(4):1221–1238.
[4] C¸ ınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., and Safaei, B. (2020). Machine learning in predictive maintenance towards sustain-able smart manufacturing in industry 4.0. Sustainabil- ity, 12(19):1–42.
[5] Davies, A. (2012). Handbook of Condition Monitoring: Techniques and Methodology. Springer Science & Business Media.
[6] Dogan, A. and Birant, D. (2021). Machine learning and data mining in manufacturing. Expert Systems with Application, 166:1–45.
[7] Erlach, K. (2013). Value Stream Design. Springer. Espinoza Sep´ulveda, N. F. and Sinha, J. K. (2021). Blind application of developed smart vibration-based ma- chine learning (svml) model for machine faults diag- nosis to different machine conditions. Journal of Vi- bration Engineering & Technologies, 9(4):587–596.
[8] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. The MIT Press. Griffiths, I. (2020). Automated condition monitoring us- ing artificial intelligence. In International Conference on Vision, Image and Signal Processing, pages 1–7. ACM. part 3-3: Application guide – life cycle costing. Stan- dard, IEC.

eISSN: 1303-5150