ISSN 2683-345X
journal homepage: http://ijiemjournal.uns.ac.rs/

International Journal of Industrial Engineering and Management

2026 / ARTICLE-IN-PRESS / 413

Original research article

Optimizing Safety Logic Device Selection with Machine Learning: A Classifier-Based Comparison

K. Stibor[a] 0009-0004-6240-5625, R. Štohl[a,*] 0000-0003-0705-5488,
L Štohlová Putnová[b] 0000-0002-9239-3133

[a] Brno University of Technology, Department of Control and Instrumentation, Brno, Czech Republic;
[b] Mendel University in Brno, Department of Animal Morphology, Physiology and Genetics, Brno, Czech Republic

ABSTRACT

In Industry 4.0 manufacturing, selecting appropriate safety logic devices during machinery risk assessment is critical yet complex. This study applies supervised machine learning classification to predict safety logic device categories. We assembled an expert-informed dataset of 306 machine cases, each described by safety-related parameters and labelled with one of four target classes (Relay, CR30, GMX, GLX). Using the WEKA toolkit, we evaluated 32 classifiers across four families (rule-based, instance-based, neural network, and tree/forest) using training-set evaluation and 5- and 10-fold cross-validation. On the holdout training data, classifiers achieved very high accuracy (average 97.1%), but cross-validation accuracies dropped to only 58–59%, indicating overfitting. Tree/forest ensembles (Random Forest, Random Tree, OptimizedForest) performed best overall (95.9% holdout test, 69.1% 5-fold cross-validation, 68.8% 10-fold cross-validation) compared to rule-based, instance-based, and multi-layer perceptron models. These results suggest that machine learning can effectively guide safety device selection, potentially reducing design time and cost in industrial safety engineering, while highlighting the need for expert oversight and larger datasets.

ARTICLE INFO

Article history:
Received August 13, 2025
Revised January 20, 2026
Accepted February 24, 2026
Published online May 09, 2026

Keywords:
Machine learning;
Industrial safety;
Classification;
Safety logic devices;
Industry 4.0

*Corresponding author:
Radek Štohl
stohl@vutbr.cz

1. Introduction

Machine learning (ML) is widely recognised as one of the key enabling technologies of Industry 4.0 and has become an integral component of data-driven manufacturing systems [1]-[6]. In industrial practice, ML techniques are commonly applied in areas such as predictive maintenance, anomaly detection, fault diagnosis, and process optimisation, where they contribute to improved efficiency and more informed decision-making [5], [6]. As illustrated in Figure 1, machine learning often forms the analytical backbone of modern cyber-physical production systems. Despite its widespread adoption in manufacturing and automation, the systematic application of ML within machinery safety engineering remains comparatively limited [7], [8].

Industrial machinery is inherently associated with safety risks, and accidents may lead to consequences ranging from minor injuries to severe or fatal harm [9], [10]. Within the framework of EN ISO 12100:2010, risk is defined as the combination of the probability of occurrence of harm and the severity of that harm.

Published by the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia.

https://doi.org/10.24867/IJIEM-413

In practice, estimating these components is challenging, as technical system characteristics interact with human behaviour and organisational conditions [10]. For this reason, machinery safety in Europe relies on structured risk assessment procedures supported by a framework of harmonised standards, including EN ISO 12100:2010, EN ISO 13849-1:2023, and EN IEC 62061:2021 [11]. Compliance with the Machinery Directive 2006/42/EC and its successor, Regulation (EU) 2023/1230, is commonly demonstrated through the application of harmonised standards.

Safety functions are typically realised through combinations of input devices, a logic subsystem, and output devices [12]. The logic subsystem plays a central role, as it processes safety-related signals and determines the appropriate system response. The selection of a suitable safety logic device (from simple safety relays to modular safety PLC systems) depends on multiple interacting factors, such as the required Performance Level (PL), the number and complexity of safety functions, the machine configuration, and the number of safety-related inputs and outputs. A representative structure of a safety function is shown in Figure 2. In practice, these decisions are rarely straightforward and often require balancing technical, economic, and organisational considerations, particularly in non-standard operating modes such as setup or maintenance [13].

Although recent studies indicate that artificial intelligence and machine learning may support safety-related engineering decisions by enabling more systematic and data-driven approaches [14], [15], their use in safety-critical contexts also raises important challenges related to interpretability, transparency, and regulatory acceptance [7]. To date, however, no published research has specifically addressed the use of supervised machine learning methods to predict categories of safety logic devices based on standardised safety parameters.

This article addresses this gap by applying supervised machine learning classification techniques to a dataset of industrial machinery cases labelled by safety engineering experts [16], [17]. The aim is to support engineers during the early stages of safety-oriented machine design by providing data-driven guidance for the selection of suitable safety logic devices, while maintaining full compliance with established safety standards.

2. Methodology

2.1. Dataset

The dataset used in this study consists of 306 industrial machinery cases that were evaluated and labelled by experienced safety engineering professionals. Each case represents a real industrial machine or production system that has undergone a formal risk assessment in accordance with applicable machinery

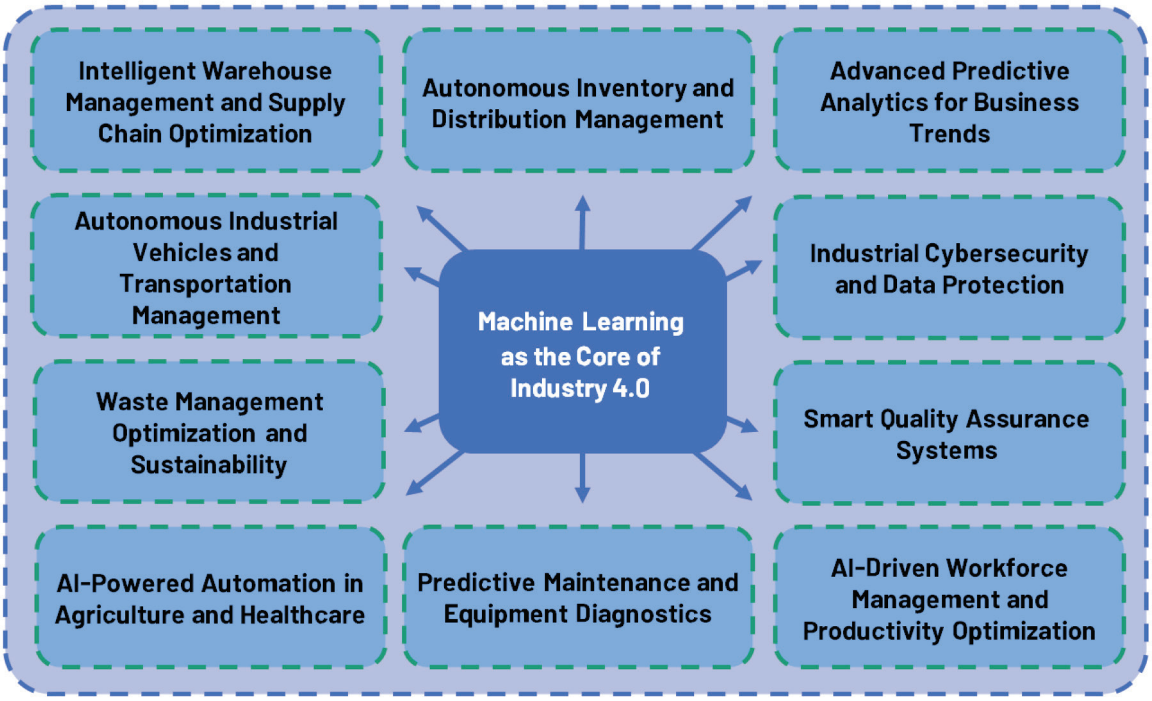


Figure 1. Machine learning as the Core of Industry 4.0

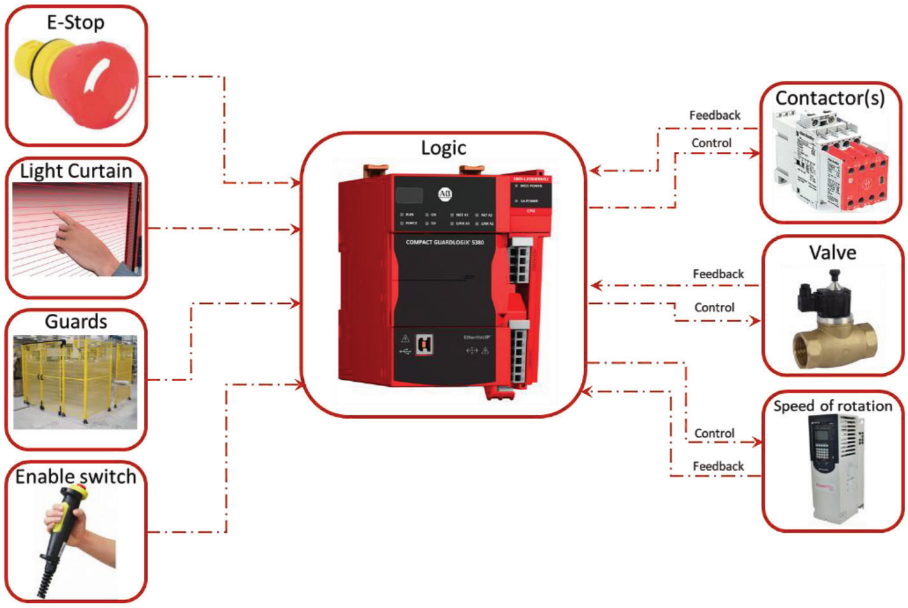


Figure 2. A logical safety functions diagram

safety standards. Each machine is described using a set of safety-relevant attributes that reflect the practical criteria typically considered during the design of safety-related control systems. An overview of all input attributes is provided in Table 1. The dataset includes numerical parameters, such as the number of safety functions, access points to hazardous areas, and the number of safety-related inputs and outputs. In addition, categorical and Boolean attributes are considered, including the required Performance Level according to EN ISO 13849-1:2023, the machine configuration, the use of a safety communication network, and the presence of a human–machine interface.

The target variable represents the category of the selected safety logic device. Four solution classes are distinguished: Relay, CR30, GMX, and GLX. These classes correspond to commonly used safety logic architectures with increasing functional capabilities and system complexity. The classes are treated as nominal categories, as no generally valid ordinal relationship exists across different machine types and application contexts.

To illustrate the variability of the numerical input parameters, Table 2 summarises the minimum and maximum values observed for selected attributes. The wide parameter ranges indicate that the dataset covers both relatively simple machines with limited safety requirements and more complex production systems with extensive safety-related I/O structures.

Beyond numerical coverage, the dataset also exhibits domain-specific representativeness. The included machine cases cover a broad range of industrial applications, including standalone equipment, multi-station production lines, and assembly systems. Moreover, the selected attributes directly correspond to the key decision criteria defined in EN ISO 12100:2010 and EN ISO 13849-1:2023,

Table 1. Attributes of the safety dataset used for classification

Attribute	Description
Number of safety functions	Number of safety functions (e.g. emergency stop, guard door) integrated in the machine (integer)
Performance Level (PL)	Required safety Performance Level (EN ISO 13849-1:2023) for the machine (categorical 'a' to 'e')
Number of access points	Number of access points to dangerous zones (e.g. maintenance openings) (integer)
Number of safety inputs	Number of safety-related sensors/inputs needed (integer)
Number of safety outputs	Number of safety-related actuators/outputs needed (integer)
Machine type	Configuration of machine: standalone unit, production line, or assembly (categorical)
Communication	Use of safety network for control (Boolean)
HMI (Human–Machine Interface)	Presence of an operator interface for safety/routine control (Boolean)
Safety solution class	Safety logic device class (target): one of {Relay, CR30, GMX, GLX}

Table 2. Minimum and maximum values of selected numerical attributes

Attribute	Minimum	Maximum
Number of safety functions	2	114
Number of access points	3	314
Number of safety inputs	1	148
Number of safety outputs	1	40

ensuring that the dataset captures the principal factors influencing safety logic device selection in industrial practice. Because these features are not tied to a specific manufacturer or application domain, the dataset structure is transferable to other industrial environments that follow comparable safety engineering principles.

2.2. Classification Approach

Supervised machine learning classification aims to assign predefined class labels to unseen data instances based on patterns learned from labelled examples [18]. In this study, classification [3] is applied to predict the category of the safety logic device based on the safety-related parameters of a machine. The overall classification concept is illustrated in Figure 3.

All experiments were conducted using the WEKA software environment [19], [20]. A total of 32 classification algorithms were evaluated and grouped into four main families: rule-based classifiers [21], [22], instance-based learning methods [23]-[25], tree- and forest-based classifiers [25], [26], and multi-layer perceptron neural networks. A detailed overview of the evaluated classifiers is provided in Table S1.

Model performance was assessed using evaluation on the training dataset and k-fold cross-validation with k = 5 and k = 10 [18]. Given the limited size of the dataset, a separate hold-out test set was not employed, as this would further reduce the number of samples available for training. Cross-validation was therefore selected as a more reliable approach for estimating the generalisation capability of the classifiers. Results obtained on the training data are reported primarily to illustrate the tendency of some models to overfit the available data.

Classification performance was evaluated using overall accuracy and Cohen's kappa as the primary performance indicators. In addition, the mean absolute error (MAE) and root mean squared error (RMSE) were calculated to provide complementary insights into prediction quality.

3. Results

The performance of the evaluated classifiers was analysed using three complementary evaluation strategies: evaluation on the training data, 5-fold cross-validation, and 10-fold cross-validation. Overall classification accuracy and Cohen's kappa were used as the primary performance measures, complemented by the mean absolute error (MAE) and root mean squared error (RMSE) to quantify prediction uncertainty and class assignment dispersion. All experiments were conducted using the WEKA environment following standard supervised learning evaluation procedures [18], [19], [20].

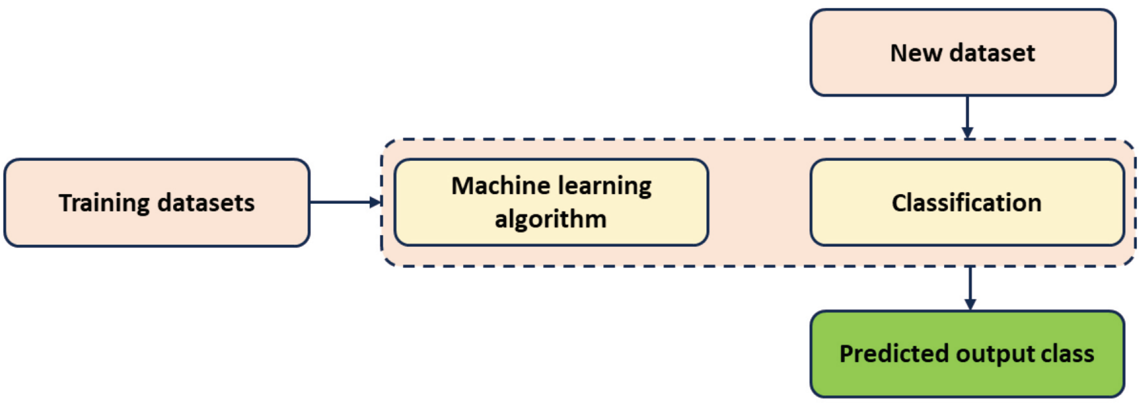


Figure 3. A logical safety functions diagram

Table 3 provides an overview of the average classification accuracy achieved by the four classifier families (instance-based learning (IBL), multi-layer perceptron (MLP) neural networks, rule-based classifiers, and tree/forest-based methods) under the different evaluation schemes. When evaluated on the training data, all classifier groups achieved very high accuracy, with an overall average of 97.06%. Instance-based learning and MLP models frequently reached perfect or near-perfect training accuracy, indicating that these algorithms were capable of closely fitting the available dataset.

However, as shown in Table 3 and illustrated in Figure 4, this high apparent performance did not translate to unseen data. Under 5-fold and 10-fold cross-validation, the average classification accuracy dropped markedly to 58.35% and 59.29%, respectively. This pronounced discrepancy between training and cross-validation results clearly indicates the presence of overfitting, which was consistently observed across all classifier families. The effect is further visualised in Figure 4, where the contrast between training and cross-validated performance is evident for all algorithm groups.

A more detailed comparison reveals substantial differences in generalisation capability between the classifier families. Tree- and forest-based classifiers achieved the highest and most stable cross-validation performance, with average accuracies of 69.12% for 5-fold cross-validation and 68.75% for 10-fold cross-validation. Rule-based classifiers ranked second, followed by MLP neural networks, whereas instance-based learning methods exhibited the weakest generalisation performance. These trends are summarised in Figure 4 and further detailed at the level of individual classifiers in Figure 5.

Cohen's kappa statistics follow the same overall pattern as classification accuracy. While kappa values were close to 1.0 for training-set evaluation, indicating almost perfect agreement, they decreased substantially under cross-validation. The highest kappa values in both 5-fold and 10-fold cross-validation were con-

Table 3. A typical machine learning approach

Group of classifiers	Evaluate on training data	5-fold cross validation	10-fold cross validation
Instance-based Learning	100.00%	45.59%	49.26%
MultiLayer Perceptron	99.02%	52.94%	53.92%
Rules	91.91%	61.76%	62.50%
Tree/Forest	95.96%	69.12%	68.75%

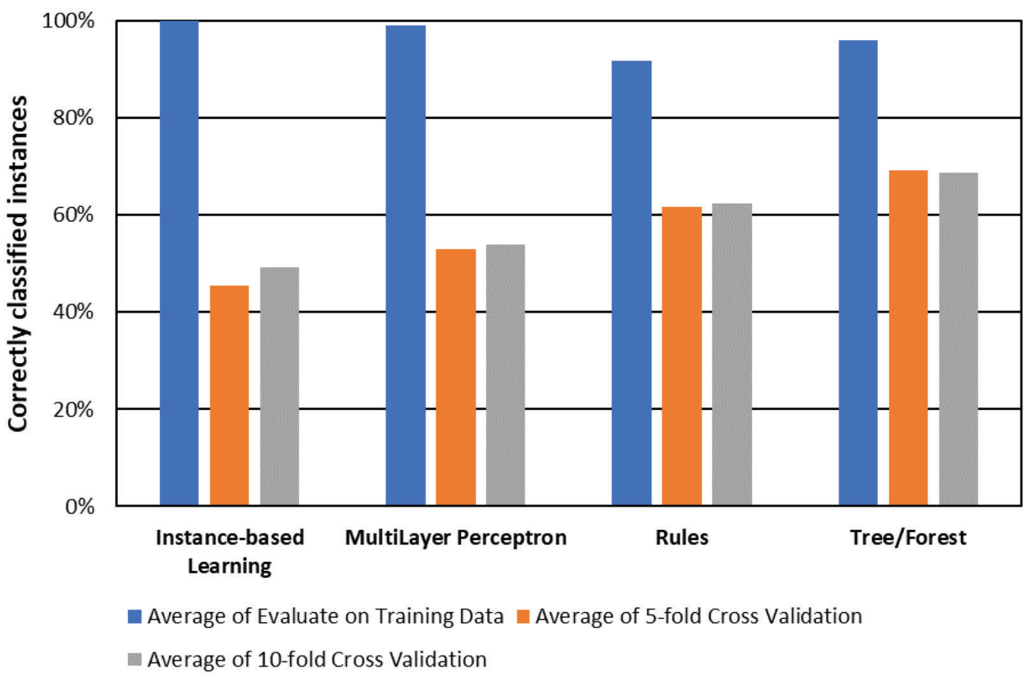


Figure 4. The average performance results obtained from the different classifier groups

sistently achieved by tree- and forest-based classifiers, confirming their superior agreement beyond chance and more reliable predictive behaviour.

Instance-based learning algorithms demonstrated the strongest tendency toward overfitting. Most IBL variants, including IB1 and IB1LG, achieved 100% training accuracy but performed poorly under cross-validation, with accuracies below 50%. The KStar algorithm represented a partial exception, reaching 61.76% accuracy in 5-fold cross-validation and 67.65% in 10-fold cross-validation, accompanied by moderate kappa values. Nevertheless, the overall performance of instance-based methods remained inferior to that of tree- and forest-based approaches.

Multi-layer perceptron models also achieved near-perfect training accuracy but showed only moderate cross-validation performance. Among the tested configurations, the MLP with eight hidden neurons (MLP8) achieved the best generalisation capability, with accuracies of 67.65% (5-fold CV) and 64.71% (10-fold CV). Other MLP variants performed less consistently, while the WiSARD network produced results close to random classification. These findings suggest that, despite their expressive capacity, neural networks require larger datasets to generalise effectively in heterogeneous safety-engineering classification tasks [18], [27].

Tree- and forest-based classifiers consistently outperformed all other algorithm families. Ensemble methods such as Random Forest, Random Tree, and OptimizedForest achieved perfect training accuracy while maintaining the highest cross-validation performance. Random Forest reached 76.47% accuracy in both 5-fold and 10-fold cross-validation, whereas OptimizedForest achieved up to 79.41% accuracy. NBTree also demonstrated robust performance, with approximately 70.6% accuracy in both cross-validation settings. In contrast, single-tree models such as J48 and BFTree exhibited noticeably lower generalisation performance. These results are consistent with previous studies highlighting the robustness of ensemble tree methods for complex classification problems involving mixed numerical and categorical attributes [28]-[30].

Figure 5 presents the cross-validation accuracy of all individual classifiers, enabling direct comparison across algorithm types. The figure clearly highlights the dominance of tree- and forest-based classifiers, particularly Random Forest, OptimizedForest, and NBTree. Rule-based classifiers and selected MLP

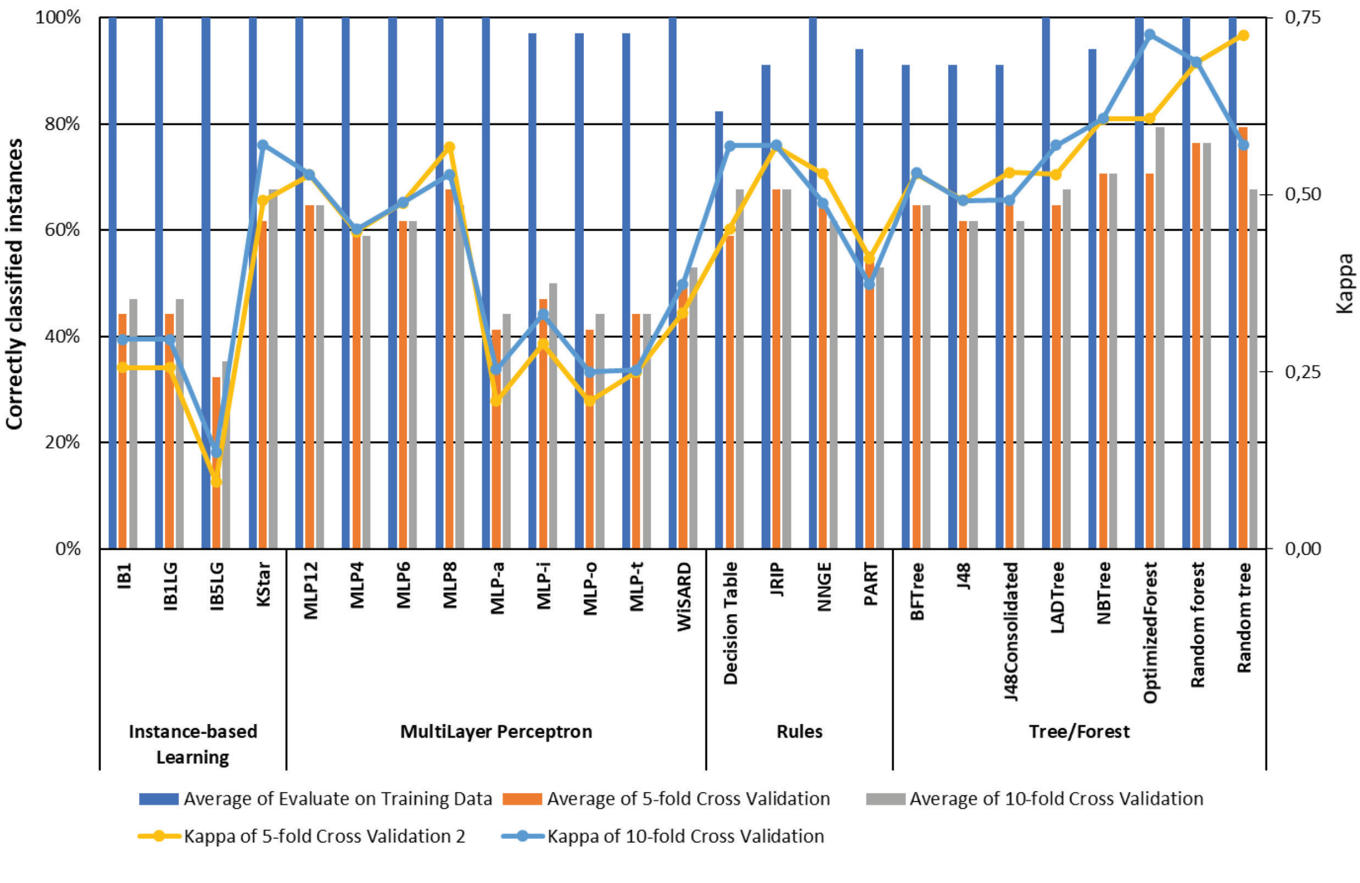


Figure 5. The performance results obtained from the different classifiers

configurations occupy an intermediate performance range, while most instance-based learners perform substantially worse [31].

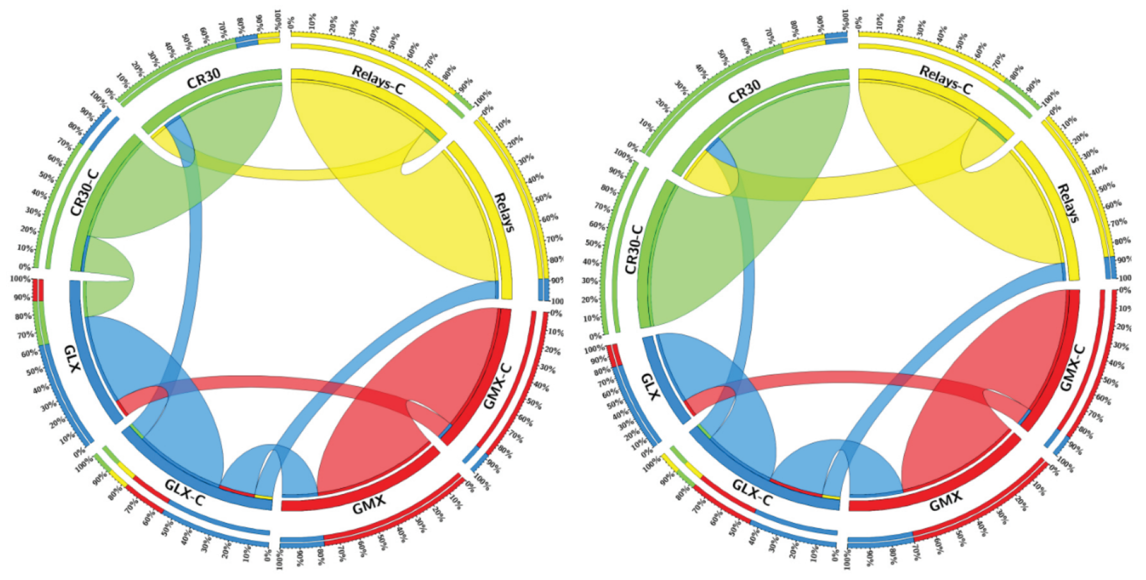
Table 4 provides a detailed breakdown of classifier performance, including Cohen's kappa, MAE, and RMSE values for each evaluation scheme. The results confirm that ensemble tree-based methods achieved the most favourable balance between accuracy and error-based metrics, indicating not only higher correctness but also lower prediction uncertainty across safety logic device classes.

To further analyse prediction behaviour at the class level, confusion matrices were examined for selected high-performing classifiers. Figure 6 shows the confusion matrix of the Random Forest classifier, which achieved the best overall performance among all evaluated models. The corresponding Circos plot [27] indicates that misclassifications predominantly occurred between safety logic device categories with comparable functional characteristics, such as CR30 and GMX, rather than between structurally dissimilar solutions. Similar class-level misclassification patterns were observed for other tree- and forest-based classifiers, as shown in Figures S1–S3.

Overall, the results demonstrate that while many classifiers are capable of fitting the training data extremely well, only a subset, primarily ensemble tree-based methods, exhibits sufficient robustness and generalisation capability to be considered reliable for safety-related decision support.

Table 4. The classifier performance rates (MAE - Mean absolute error, RMSE - Root mean squared error)

Classifier group	Evaluate on training data				5-fold cross validation				10-fold cross validation			
	Correctly classified	Kappa statistics	MAE	RMSE	Correctly classified	Kappa statistics	MAE	RMSE	Correctly classified	Kappa statistics	MAE	RMSE
Rules												
Decision Table	82.35%	0.7655	0.2180	0.2930	58.82%	0.4522	0.2874	0.3733	67.65%	0.5691	0.2804	0.3622
JRIP	91.18%	0.8826	0.0642	0.1791	67.65%	0.5686	0.1737	0.3968	67.65%	0.5701	0.1778	0.3721
NNGE	100.00%	1.0000	0.0000	0.0000	64.71%	0.5294	0.1765	0.4201	61.76%	0.4884	0.1912	0.4372
PART	94.12%	0.9217	0.0482	0.1552	55.88%	0.4097	0.2147	0.4324	52.94%	0.3733	0.2315	0.4402
Instance-based learning												
IB1	100.00%	1.0000	0.0372	0.0438	44.12%	0.2558	0.2890	0.4972	47.06%	0.2957	0.2749	0.4862
IB1LG	100.00%	1.0000	0.0808	0.0949	44.12%	0.2558	0.3001	0.4648	47.06%	0.2957	0.4565	0.2870
IB5LG	100.00%	1.0000	0.0808	0.0949	32.35%	0.0949	0.3160	0.4488	35.29%	0.1363	0.3009	0.4373
KStar	100.00%	1.0000	0.0008	0.0019	61.76%	0.4914	0.1794	0.3697	67.65%	0.5706	0.1744	0.3585
MultiLayer perceptron												
MLP12	100.00%	1.0000	0.1187	0.1645	64.71%	0.5278	0.2587	0.3759	64.71%	0.5283	0.2578	0.3907
MLP4	100.00%	1.0000	0.1385	0.1812	58.82%	0.4478	0.2646	0.3766	58.82%	0.4510	0.2765	0.3926
MLP6	100.00%	1.0000	0.1305	0.1745	61.76%	0.4884	0.2632	0.3804	61.76%	0.4890	0.2608	0.3844
MLP8	100.00%	1.0000	0.1240	0.1675	67.65%	0.5676	0.2598	0.3805	64.71%	0.5283	0.2638	0.3887
MLP-a	100.00%	1.0000	0.0696	0.1289	41.18%	0.2084	0.2937	0.4659	44.12%	0.2532	0.2833	0.4576
MLP-i	97.06%	0.9607	0.0649	0.1275	47.06%	0.2892	0.2769	0.4436	50.00%	0.3310	0.2554	0.4323
MLP-o	97.06%	0.9607	0.0726	0.1335	41.18%	0.2084	0.2905	0.4547	44.12%	0.2497	0.2862	0.4570
MLP-t	97.06%	0.9607	0.0620	0.1233	44.12%	0.2497	0.2836	0.4469	44.12%	0.2523	0.2700	0.4439
WiSARD	100.00%	1.0000	0.3547	0.4732	50.00%	0.3333	0.4134	0.5015	52.94%	0.3740	0.4287	0.5146
Tree/Forest												
BFTree	91.18%	0.8826	0.0723	0.1901	64.71%	0.5294	0.1976	0.4039	64.71%	0.5305	0.1894	0.3705
J48	91.18%	0.8826	0.0723	0.1901	61.76%	0.4920	0.2114	0.4142	61.76%	0.4914	0.2095	0.4155
J48 Consolidated	91.18%	0.8826	0.0723	0.1901	64.71%	0.5316	0.1979	0.4049	61.76%	0.4925	0.2146	0.4202
LADTree	100.00%	1.0000	0.0000	0.0000	64.71%	0.5289	0.1780	0.4088	67.65%	0.5696	0.1551	0.3865
NBTree	94.12%	0.9217	0.0701	0.1524	70.59%	0.6069	0.1595	0.2841	70.59%	0.6074	0.1646	0.3020
OptimizedForest	100.00%	1.0000	0.0619	0.1080	70.59%	0.6078	0.1891	0.3064	79.41%	0.7264	0.1735	0.2949
Random Forest	100.00%	1.0000	0.0656	0.1090	76.47%	0.6863	0.1882	0.3044	76.47%	0.6877	0.1767	0.2958
Random Tree	100.00%	1.0000	0.0000	0.0000	79.41%	0.7252	0.1157	0.3291	67.65%	0.5706	0.1618	0.4022



a) 5-fold cross validation (accuracy 76.47 %)

b) 10-fold cross validation (accuracy 76.47 %)

Figure 6. The Circos plot displaying the confusion matrix of the safety solution classes and summarizing the dominant prediction power of the Random Forest classifier within the Tree/Forest group

4. Discussion

The results demonstrate that supervised machine learning can support the selection of safety logic devices when trained on safety-relevant parameters derived from real industrial practice. At the same time, the findings highlight the limitations associated with applying flexible classification models to relatively small datasets, a challenge that has been widely reported in safety-related and industrial machine learning applications [18], [32]-[35]. The pronounced difference between training-set performance and cross-validation results (Table 3, Figure 4) indicates substantial overfitting across all evaluated classifier families. This behaviour suggests that the available dataset does not fully capture the variability of safety-related design scenarios encountered in industrial machinery. In practical terms, this finding underlines the importance of careful model validation and reinforces the need for sufficiently large and diverse datasets when machine learning methods are applied in safety engineering contexts. Among the evaluated classifier families, tree- and forest-based methods consistently achieved the most robust generalisation performance (Table 3, Figure 5). This observation is in line with previous studies reporting strong performance of ensemble learning techniques in heterogeneous classification tasks [26], [35], [36]. From an engineering perspective, this result is also intuitive, as the selection of safety logic devices typically follows hierarchical and rule-like decision structures that are well captured by decision trees and their ensembles.

The analysis of confusion matrices provides additional insight into the practical relevance of the proposed approach. As shown in Figure 6 and Figures S1–S3, misclassifications predominantly occurred between safety logic device categories with comparable functional capabilities. Importantly, the models rarely confused fundamentally different solution types, such as simple relay-based architectures and highly modular safety PLC systems [37]. This behaviour reflects real-world engineering trade-offs and suggests that the classification errors are generally plausible from a practical design perspective. Despite these encouraging results, the findings do not suggest that machine learning should replace expert judgement in machinery safety engineering. Safety-related decisions are subject to regulatory requirements, normative constraints, and context-specific considerations that cannot be fully captured by data-driven models alone [7], [12]. Instead, the proposed approach should be understood as a decision-support tool that complements established risk assessment and safety design procedures. In this role, machine learning can contribute to improved consistency and efficiency, particularly during early design stages or when dealing with complex machine configurations.

Finally, it should be noted that the current study focuses exclusively on static safety parameters derived from risk assessment documentation. Future work may benefit from integrating additional information sources, such as operational data, sensor signals, or human–machine interaction data, which have been shown to enhance safety-related decision-making in smart manufacturing environments [9], [15].

5. Conclusion

This study explored the application of supervised machine learning methods for predicting categories of safety logic devices based on standardised safety parameters derived from industrial machinery risk assessments. A comprehensive comparison of 32 classification algorithms showed that tree- and forest-based methods achieved the most reliable generalisation performance across different evaluation schemes. The results indicate that ensemble tree-based classifiers, particularly Random Forest, Random Tree, and OptimizedForest, are well suited to the considered task, as they provide a favourable balance between classification accuracy and robustness. At the same time, the observed performance gap between training-set evaluation and cross-validation highlights the limitations imposed by the size and diversity of the available dataset.

From a practical perspective, the proposed approach demonstrates the potential of machine learning to support engineers during the selection of safety logic devices, especially in early design phases or when dealing with complex machine configurations. When used as a decision-support tool, such methods may contribute to reduced design effort and improved consistency across projects, while remaining fully aligned with established machinery safety standards.

Future research should focus on extending the dataset to include a broader range of machinery types and safety scenarios, thereby improving model robustness and generalisation. In addition, hybrid approaches that combine data-driven models with expert rules and normative constraints may further enhance the applicability of machine learning in safety-critical engineering contexts.

Acknowledgement

The authors thank the editor and co-editor, respectively, for their time, effort, and attention to the manuscript; further, they acknowledge comments and recommendations proposed by the anonymous reviewers, as these have markedly helped to refine the manuscript.

Funding

This work was supported by the Internal Science Fund of Brno University of Technology project "Research on advanced methods and technologies in cybernetics, robotics, artificial intelligence, automation and measurement" [grant agreement number FEKT-S-23-8451].

References

[1] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: Advantages, challenges, and applications," Prod. Manuf. Res., vol. 4, no. 1, pp. 23–45, 2016, doi: 10.1080/21693277.2016.1192517.

[2] V. Alcácer and V. Cruz-Machado, "Scanning Industry 4.0: A literature review on technologies for manufacturing systems," Eng. Sci. Technol., Int. J., vol. 22, no. 3, pp. 899–919, 2019, doi: 10.1016/j.jestch.2019.01.006.

[3] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," Sensors, vol. 18, no. 8, p. 2674, 2018, doi: 10.3390/s18082674.

[4] M. S. Rahman, T. Ghosh, N. F. Aurna, M. S. Kaiser, M. Anannya, and A. S. M. S. Hosen, "Machine learning and internet of things in industry 4.0: A review," Meas. Sensors, vol. 28, 2023, doi: 10.1016/j.measen.2023.100822.

[5] T. Chen et al., "Machine learning in manufacturing towards Industry 4.0: From 'For Now' to 'Four-Know,'" Appl. Sci., vol. 13, p. 1903, 2023, doi: 10.3390/app13031903.

[6] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," Expert Syst. Appl., vol. 175, p. 114820, 2021, doi: 10.1016/j.eswa.2021.114820.

[7] S. Anastasi, M. Madonna, and L. Monica, "Implications of embedded artificial intelligence–machine learning on safety of machinery," Procedia Comput. Sci., vol. 180, pp. 338–343, 2021, doi: 10.1016/j.procs.2021.01.171.

[8] N. Becker, P. Junginger, L. Martinez, D. Krupka, and L. Beining, AI at Work – Mitigating Safety and Discriminatory Risk with Technical Standards. Berlin, Germany: Gesellschaft für Informatik, 2021.

[9] S. R. Addula and A. Tyagi, "Future of computer vision and industrial robotics in smart manufacturing," in Artificial Intelligence-Enabled Digital Twin for Smart Manufacturing. Hoboken, NJ, USA: Wiley, 2024, pp. 505–539, doi: 10.1002/9781394303601.ch22.

[10] Y. Chinniah, B. Aucourt, and R. Bourbonnière, "Safety of industrial operations: Development of the hierarchical enumeration of risks (HER) table," in Proc. Annu. Reliab. Maint. Symp. (RAMS), 2017, pp. 1–6.

[11] S. Jocelyn, M. S. Ouali, and Y. Chinniah, "Estimation of probability of harm in safety of machinery using an investigation systemic approach and Logical Analysis of Data," Saf. Sci., vol. 105, pp. 32–45, 2018, doi: 10.1016/j.ssci.2018.01.018.

[12] ISO/IEC Guide 51, Safety Aspects—Guidelines for Their Inclusion in Standards, 3rd ed., 2014.

[13] J. Saenz, R. Behrens, E. Schulenburg et al., "Methods for considering safety in design of robotics applications

featuring human-robot collaboration," Int. J. Adv. Manuf. Technol., vol. 107, pp. 2313–2331, 2020, doi: 10.1007/s00170-020-05076-5.
[14] S. Kim, J. Lee, and C. Kang, "Analysis of industrial accidents causing through jamming or crushing accidental deaths in the manufacturing industry in South Korea: Focus on non-routine work on machinery," Saf. Sci., vol. 133, 2021, doi: 10.1016/j.ssci.2020.104998.
[15] L. Leoni, A. BahooToroody, M. M. Abaei, A. Cantini, F. BahooToroody, and F. De Carlo, "Machine learning and deep learning for safety applications: Investigating the intellectual structure and the temporal evolution," Saf. Sci., vol. 170, 2024, doi: 10.1016/j.ssci.2023.106363.
[16] Z. Xu and J. H. Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," Reliab. Eng. Syst. Saf., vol. 211, p. 107530, 2021, doi: 10.1016/j.ress.2021.107530.
[17] R. Štohl and K. Stibor, "Predicting safety solutions via an artificial neural network," IFAC-PapersOnLine, vol. 52, no. 27, pp. 490–495, 2019, doi: 10.1016/j.ifacol.2019.12.711.
[18] R. Štohl and K. Stibor, "Predicting safety logic device solutions via decision trees and rules algorithms," in Proc. 21st Int. Carpathian Control Conf. (ICCC), 2020, pp. 1–7, doi: 10.1109/ICCC49264.2020.9257284.
[19] M. Awad and R. Khanna, Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. New York, NY, USA: Apress, 2015.
[20] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: A machine learning workbench," in Proc. 2nd New Zealand Conf. Artificial Intelligence, 1994, pp. 357–361, doi: 10.1109/ANZIIS.1994.396988.
[21] E. Frank, M. A. Hall, and I. H. Witten, The WEKA Workbench: Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Burlington, MA, USA: Morgan Kaufmann, 2016.
[22] W. W. Cohen, "Fast effective rule induction," in Proc. 12th Int. Conf. Machine Learning, 1995, pp. 115–123, doi: 10.1016/B978-1-55860-377-6.50023-2.
[23] M. Brent, Instance-Based Learning: Nearest Neighbor with Generalization, Master's thesis, Univ. of Waikato, Hamilton, New Zealand, 1995.
[24] J. G. Cleary and L. E. Trigg, "K*: An instance-based learner using an entropic distance measure," in Proc. 12th Int. Conf. Machine Learning, 1995, pp. 108–114.
[25] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," Mach. Learn., vol. 6, no. 1, pp. 37–66, 1991, doi: 10.1007/BF00153759.
[26] J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[27] L. Breiman, "Random forests," Mach. Learn., vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[28] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra, "Circos: An information aesthetic for comparative genomics," Genome Res., vol. 19, no. 9, pp. 1639–1645, 2009, doi: 10.1101/gr.092759.109.
[29] C. Anton, A. Avram, O. Matei, L. Andreica, and B. Văduva, "Advancements in machine learning algorithms for precision crop yield prediction: A comprehensive review with focus on European Union," in Proc. 19th Int. Conf. Soft Comput. Models Ind. Environ. Appl. (SOCO 2024), H. Quintián et al., Eds. Cham, Switzerland: Springer, 2025, pp. xx–xx, doi: 10.1007/978-3-031-75010-6_18.
[30] S. O. Araújo, R. S. Peres, J. C. Ramalho, F. Lidon, and J. Barata, "Machine learning applications in agriculture: Current trends, challenges, and future perspectives," Agronomy, vol. 13, 2023, doi: 10.3390/agronomy13122976.
[31] D. Ferreira, P. Corista, J. Gião, S. Ghimire, J. Sarraipa, and R. Jardim-Gonçalves, "Towards smart agriculture using FIWARE enablers," in Proc. Int. Conf. Eng., Technol. Innov. (ICE/ITMC), Madeira, Portugal, 2017, pp. 1544–1551, doi: 10.1109/ICE.2017.8280066.
[32] N. Nainggolan, E. Maghsoudlou, B. M. AlWadi, F. Atamurotov, M. E. Kosov, and W. Putra, "Advancements in optimization for automotive manufacturing: Hybrid approaches and machine learning," Int. J. Ind. Eng. Manag., vol. 15, no. 3, pp. 254–263, Sep. 2024, doi: 10.24867/IJIEM-2024-3-361.
[33] R. Kallenbach, C. Reichert, F. Hieronymi, and R. Nicodemus, "Mobility 4.0—On the road toward mobility services," ATZ Worldw., vol. 121, pp. 64–69, 2019, doi: 10.1007/s38311-018-0188-3.
[34] K. Gąsiorek, "Key competences for Transport 4.0—Educators' and practitioners' opinions," Open Eng., vol. 12, no. 1, pp. 51–61, 2022, doi: 10.1515/eng-2022-0009.
[35] Y. Chinniah, B. Aucourt, and R. Bourbonnière, "Methods for considering safety in design of robotics applications featuring human-robot collaboration," Int. J. Adv. Manuf. Technol., vol. 107, pp. 2313–2331, 2020, doi: 10.1007/s00170-020-05076-5.
[36] S. Wang, M. Li, L. Wang, X. Han, and J. Li, "Regional disparities and evolutionary trends in physician resources in China: A deep learning and text mining approach," Teh. Vjesn., vol. 32, no. 5, pp. 1690–1704, 2025, doi: 10.17559/TV-20240721001871.
[37] G. Biau and E. Scornet, "A random forest guided tour," Test, vol. 25, pp. 197–227, 2016, doi: 10.1007/s11749-016-0481-7.

Appendix

Table S1. Various classification methods for individual safety logic device solutions

Classifier group	Description	References
Rules		
Decision Table	This algorithm represents a class for constructing and utilizing a simple decision table majority classifier.	Kohavi 1995
JRIP	It implements a propositional rule learner known as Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which is an optimized version of incremental reduced error pruning.	Cohen 1995
NNGE	The Non-Nested Generalized Exemplars algorithm performs a generalization task by combining samples, creating hyperrectangles in the feature space that present conjunctive rules with internal disjunctions.	Brent 1995
PART	This tool integrates the C4.5 and RIPPER methods, aiming to mitigate the issues associated with each approach.	Frank and Witten 1998
Instance-based learning		
IB*k*	This algorithm is a general *k*-nearest-neighbour classifier. When $k = 1$, the new point is classified based on the nearest neighbour's data point. When $k = 5$, the class of the new point is determined by the five nearest neighbours.	Aha and Kibler 1991, Larose and Larose 2014, Mohamed 2017
IB*k*LG	The algorithm allows for weighting using Log and Gaussian kernels, as well as various distance measures and related factors, such as inverse or similarity-based variations.	Sreenivasamurthy and Frank 2015
KStar	It employs entropy-based distance metrics to assess similarity between instances, making it more adaptable than other instance-based methods for handling different data distributions.	Cleary and Trigg 1995
Tree/Forest		
J48	The open-source Java implementation of the C4.5 algorithm in WEKA.	Quinlan 1993
J48 Consolidated	The algorithm generates either a pruned or unpruned C4.5 decision tree. It uses a resampling method to generate samples for the consolidation process.	Pérez et al. 2007, Ibarguren et al. 2015
Random Tree	The ensemble learning algorithm creates multiple individual learners and employs bagging to form a random set of data for constructing decision trees.	Mishra and Ratha 2016
Random Forest	The core function involves constructing numerous decision trees during training and outputting the class.	Breiman 2001, Biau and Scornet 2016, Mishra and Ratha 2016
BFTree	The algorithm uses binary splits for both nominal and numeric attributes. For missing values, it employs the method of 'fractional' instances.	Friedman et al. 2000, Shi 2007
LADTree	This function allows for generating a multi-class alternating decision tree using the LogitBoost strategy.	Holmes et al. 2001
NBTree	The algorithm produces decision trees with naive Bayes classifiers at the leaves.	Kohavi 1996
CSForest	It includes an implementation of the cost-sensitive decision forest algorithm.	Islam and Giggins 2011, Siers and Islam 2015
JCHAIDStar	A decision tree is generated based on the CHAID algorithm, which uses chi-squared (X^2) to measure the correlation between attributes and the class, handling only discrete variables while distinguishing between nominal and ordinal types.	Kass 1980, Ibarguren et al. 2016
OptimizedForest	Additionally, it implements the optimal sub-forest algorithm, building a decision forest to determine an optimal sub-forest using a genetic algorithm.	Adnan and Islam 2016

Classifier group	Description	References
MultiLayer perceptron		
MLP*n*	This tool trains a multilayer perceptron with one hidden layer using WEKA's optimization class, minimizing the given loss function plus a quadratic penalty with the Broyden–Fletcher–Goldfarb–Shanno algorithm. The number of neurons in the hidden layer is defined as n (n = 4, 8, 12).	Nawi et al. 2006
MLPCS-*x*	The classifier utilizes backpropagation to classify instances. It includes two important corrections and five extensions to the existing MLP. The number of neurons in one hidden layer is defined through an x code (x = 4, 8, 12).	Tu et al. 2010
MLP-*X*	This classifier uses backpropagation to train a multilayer perceptron to classify instances. The nodes in this network are all sigmoid, and training is conducted via the backpropagation procedure. The number of neurons in one hidden layer is defined through an X code (X = a, i, o, t).	Li et al. 2012
WiSARD	This implementation adds a data pre-processing filter method to the process. Generally, the Wilkes, Stonham, and Aleksander Recognition Device (WiSARD) is a neural model with training and classification capabilities.	De Gregorio and Giordano 2018

References from Table S1

Kohavi, R. (1995). The power of decision tables. Machine Learning: ECML-95. Lecture Notes in Computer Science. Springer. Berlin. Heidelberg. 174-189. https://doi.org/10.1007/3-540-59286-5_57.

Cohen, W. W. (1995). Fast effective rule induction". Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA.

Brent, M. (1995). Instance-based learning: Nearest neighbor with generalization. Master's thesis, University of Waikato, Hamilton, New Zealand.

Frank, E., Witten, I. H. (1998). Generating accurate rule sets without global optimization. Proc Fifteenth International Conference on Machine Learning. San Francisco, CA.

Aha, D. W., Kibler, D., Albert, M. K. (1991). Instance-based learning algorithms. Machine Learning 6(1). 37–66.

Cleary, J. G., & Trigg, L. E. (1995). K*: An instance-based learner using an entropic distance measure. In Proceedings of the 12th International Conference on Machine Learning (pp. 108–114).

Larose, D. T., Larose, C. D. (2014). Imputation of Missing Data. Discovering Knowledge in Data: An Introduction to Data Mining. Wiley. 266-276. https://doi.org/10.1002/9781118874059.

Mohamed, A. E. (2017). Comparative study of four supervised machine learning techniques for classification. International Journal of Applied Science and Technology 7(2), 1-15.

Sreenivasamurthy, S., Frank, S. (2015). Efficacy of season prediction for geo-locations using geo-tagged images. 2015 IEEE International Conference on Information Reuse and Integration. 476-484.

Quinlan, R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers.

Pérez, J. M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., Martín, J. I. (2007). Combining multiple class distribution modified subsamples in a single tree. Pattern Recognition Letters 28(4), 414-422. https://doi.org/10.1016/j.patrec.2006.08.013.

Ibarguren, I., Pérez, J. M., Muguerza, J., Gurrutxaga, I., Arbelaitz, O. (2015). Coverage-based resampling: Building robust consolidated decision trees. Knowledge Based Systems 79, 51-67. https://doi.org/10.1016/j.knosys.2014.12.023.

Mishra, A. K., Ratha, B. K. (2016). Study of Random Tree and Random Forest Data Mining Algorithms for Microarray Data Analysis. International Journal on Advanced Electrical and Computer Engineering. 3(4), 5-7.

Breiman, L. (2001). Random forests. Machine Learning 45, 5–32.

Biau, G., Scornet, E. (2016). A random forest guided tour. Test 25, 197–227. https://doi.org/10.1007/s11749-016-0481-7.

Friedman, J., Hastie, T., Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. Annals of statistics. 28(2). 337-407.

Shi, H. (2007). Best-first decision tree learning. Thesis. The University of Waikato, Hamilton, New Zealand. https://hdl.handle.net/10289/2317.

Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., Hall, M. (2001). Multiclass alternating decision trees. Machine Learning: ECML 2002. 161-172. Lecture Notes in Computer Science. 2430. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36755-1_14.

Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. Second International Conference on Knowledge Discovery and Data Mining. 202-207.

Islam, M. Z., Giggins, H. (2011). Knowledge discovery through sysfor: a systematically developed forest of multiple decision trees, Proceedings of the Ninth Australasian Data Mining Conference, vol. 121, Australian Computer Society, Inc., Ballarat, Australia, 195–204.

Siers, M. J., Islam, M. Z. (2015). Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem. Information Systems 51, 62-71.

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. Journal of the Royal Statistical Society. Series C (Applied Statistics). 29(2), 119-127. http://www.jstor.org/stable/2986296.

Ibarguren, I., Lasarguren, A., Pérez, J. M., Muguerza, J., Gurrutxaga, I., Arbelaitz, O. (2016). BFPART: Best-First PART. Information Sciences 367-368, 927-952. https://doi.org/10.1016/j.ins.2016.07.023.

Adnan, M. N., Islam, M. Z. (2016). Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm. Knowledge-Based Systems. 110.

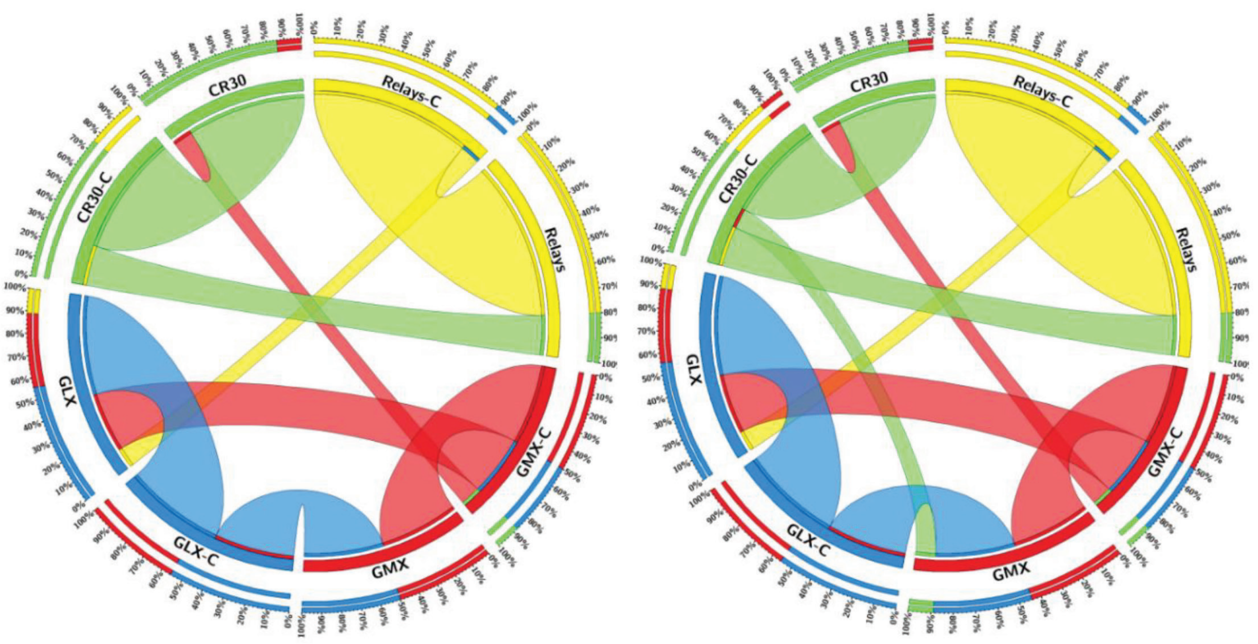
Nawi, N. M., Ransing, M. R., Ransing, R. S. (2006). An Improved Learning Algorithm Based on The Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method For Back Propagation Neural Networks. Sixth International Conference on Intelligent Systems Design and Applications. 152-157, https://doi.org/10.1109/ISDA.2006.95.

Tu, L., Fowler, B., Silver, D. L. (2010). CsMTL MLP For WEKA: Neural Network Learning with Inductive Transfer. Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010). The Florida AI Research Society. USA.

Li, J., Cheng, J. H., Shi, J. Y., Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. Advances in Computer Science and Information Engineering. 2. Springer Berlin Heidelberg. 553-558.

De Gregorio M., Giordano M. (2018). An experimental evaluation of weightless neural networks for multi-class classification, Applied Soft Computing, 72, 338-354, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2018.07.052.

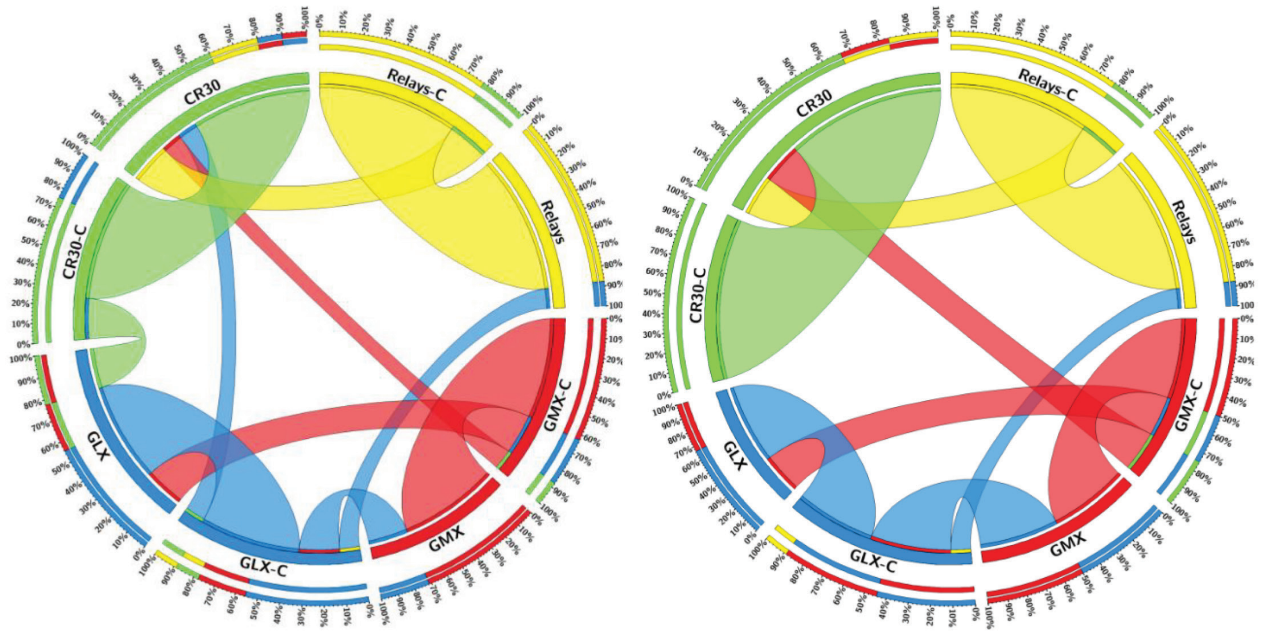
Cleary JG, Trigg LE. K*: An Instance-based Learner Using an Entropic Distance Measure. In: Prieditis A, Russell S, editors. Proceedings of the 12th International Conference on Machine Learning; 1995 Jul 9-12; Tahoe City, California. Morgan Kaufmann; 1995. p. 108-114. doi 10.1016/B978-1-55860-377-6.50022-0



a) 5-fold cross validation (accuracy 67.65 %)

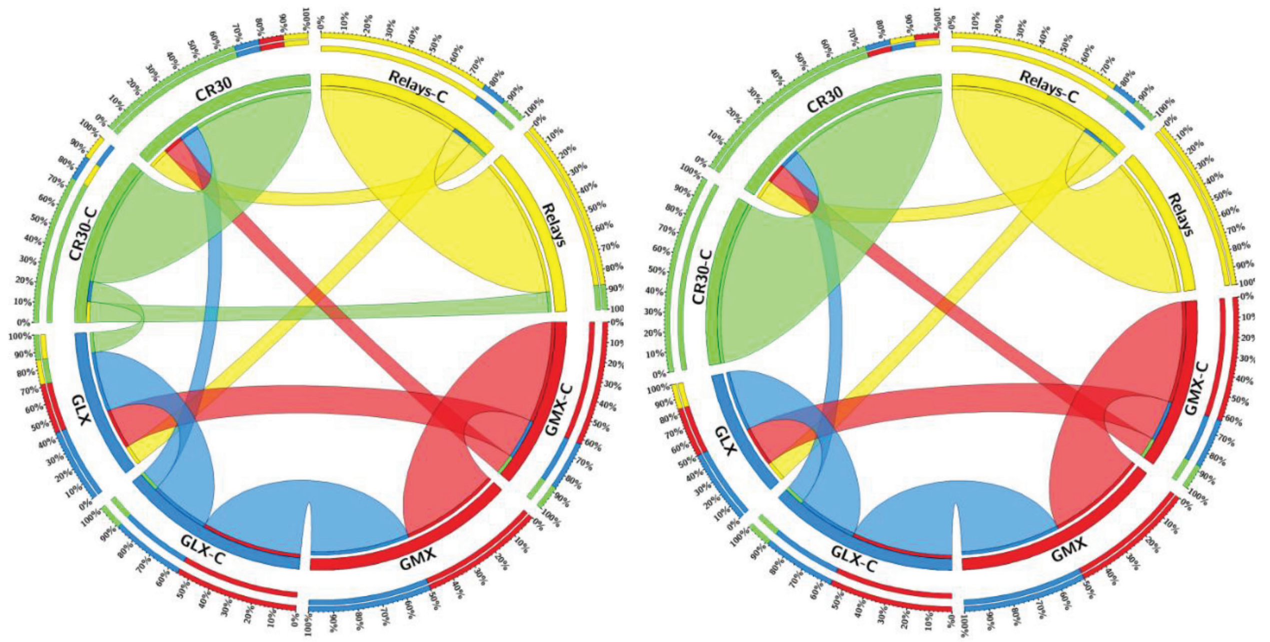
b) 10-fold cross validation (accuracy 64.71 %)

Figure S1. Prediction power of the MLP 8 classifier within the MLP group



a) 5-fold cross validation (accuracy 67.65 %) b) 10-fold cross validation (accuracy 67.65 %)

Figure S2. Prediction power of the JRIP classifier within the Rules group



a) 5-fold cross validation (accuracy 61.76 %) b) 10-fold cross validation (accuracy 67.65 %)

Figure S3. Prediction power of the KStar classifier within the IBL group