Machine Learning Versus Deep Learning for Contact Detection in Human-Robot Collaboration

Lydia N. Faraj[*], **Baraa M. Albaker**[**], **Asmaa H. Rasheed**[***]

[*] College of Engineering, Al-Iraqia University, Saba'a Abkar Complex, Baghdad, Iraq
Email: lydia.n.faraj@aliraqia.edu.iq
https://orcid.org/0009-0009-0557-4234

[**] College of Engineering, Al-Iraqia University, Saba'a Abkar Complex, Baghdad, Iraq
Email: baraamalbaker@ymail.com
https://orcid.org/0000-0002-6030-3121

[***] Electronics and Communications Department, College of Engineering, Al-Nahrain University, Baghdad, Iraq
Email: d.asmaa.mas@gmail.com
https://orcid.org/0000-0001-5589-4214

Abstract

Due to the rapid progression of Human-Robot Collaboration (HRC), ensuring safe interactions between humans and robots, contact detecting systems must be dependable and efficient. In this research, various models are tested using a contact detection dataset that includes non-contact motions, intentional interactions, and accidental collisions among others. K-Nearest Neighbors (KNN), Bagging, and Long Short-Term Memory (LSTM) networks are evaluated on their ability to classify different types of contacts. According to the findings of the experiment, it is clear that KNN and Bagging are reasonably accurate, but LSTM has surpassed both by achieving higher accuracy levels besides being better at handling temporal dependencies which are inherent in sensor data collected from dynamic human-robot interactions. The results have shown that when it comes to such kind of contact detection datasets, long short-term memory (LSTM) and other deep learning models are superior to other methods. These results show that HRC systems can be made safer and more effective by using more sophisticated neural networks. This research helps connect theory with practice by providing a foundation for the creation of collaborative robots that are not only intelligent but also safe.

Keywords- Contact detection, Deep learning, Human-robot collaboration, Machine learning, Safety.

I. INTRODUCTION

Human-Robot Collaboration (HRC) involves humans and robots working together in an integrated system that shares tasks and physical space to realize common objectives. Different collaborative tasks have been reviewed, categorized, and analyzed in a previous work [1] [2]. HRC aims to create a symbiotic environment where people and machines can co-exist harmoniously, improve productivity, and stimulate innovation [3] [4]. However, integrating robots into shared workspaces introduces significant safety risks [5]. Collaborative robots or cobots are designed to directly collaborate with humans rather than being confined within walls or protective barriers like traditional industrial robots. Due to no physical barriers in place, the probability of injury is increased through unintended contact or collisions. For instance, robot motions can be instantly stopped when it detects physical contact or breach of a safety distance as specified by compliance regulations. However, such efforts may limit the fluency and efficacy of human-robot collaborations [6] [7]. Humans are required to interact physically with robots in order to perform collaborative and coordinative tasks.
In many cooperation cases, purposeful physical contact, like leading a robot or handing over objects, is necessary for accomplishing the tasks. Workers often interface with robots to adjust parts or tools [8] [9]. Robots should be able to differentiate between non-contact motions, intentional contacts, and accidental crashes during these interactions. Therefore, it is important to identify and react effectively to these different types of physical touches if we want our human-robot collaboration systems to be meaningful and safe. Furthermore, robotic arms such as the Franka Panda, for instance, have several joints and links between these joints, as depicted in Figure 1, where a manipulator with seven rotational joints is shown [10]. Identifying the exact link of the manipulator in which the contact occurs is highly beneficial, as it allows for more precise reaction schemes to be made in the further phases of safety system design.

http://doi.org/10.58564/IJSER.3.3.2024.234

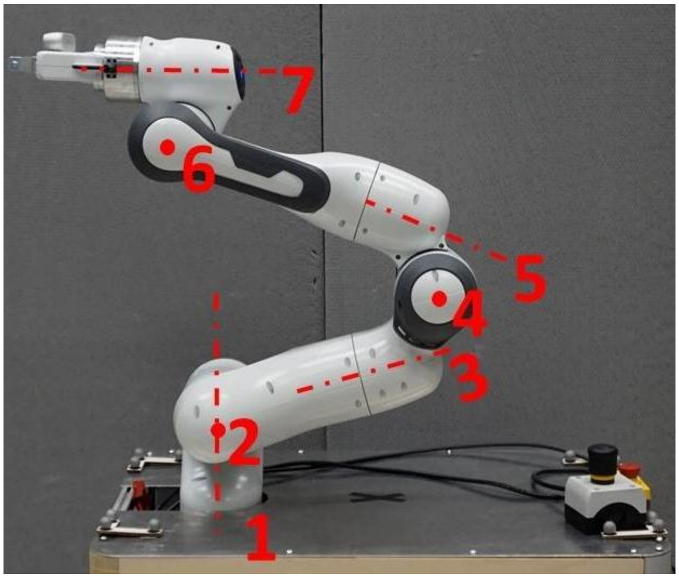

Figure 1. Seven-joint manipulator robotic rm [10]

To tackle these obstacles, sophisticated touch-sensitive contact detection systems are required to be created with the help of machine learning algorithms. This article presents the assessment of three models in classifying contact states, including the K-Nearest Neighbors (KNN), Bagging classifier, and Long Short-Term Memory (LSTM) neural network. The study aims to promote safe and efficient human-robot cooperation by differentiating between movements without contact, deliberate interaction, and collision situations, which in turn allows robots to adjust their behavior based on the changing needs of operators.

Generally, there are two types of collision detection methods in recent studies. On the one hand, model-based techniques use disturbance observers, impedance and admittance control, and force and torque sensors. However, these methods need precise dynamic parameters of robots, which is hard to get accurately. On the other hand, data-driven approaches are becoming more popular due to their practicality and success rates. These methods involve collecting data from robots and then processing it using different AI algorithms.

To boost collaboration, Salehzadeh et al. [11] have carried out an overall analysis of contemporary manufacturing applications with greater emphasis on effective communication strategies. Sharkawy and Koustoumpardis [12] give a detailed study of Human-Robot Interaction (HRI), which mainly focuses on variable admittance control, safety considerations, and future perspectives. According to their findings, adaptive control mechanisms should be able to guarantee safety without compromising efficiency in HRC environments. Detecting and differentiating various human-robot contact types, such as non-contact movements, intentional interactions, and accidental collisions, is crucial for safe collaboration. Sharkawy and Aspragathos [13] delve into collision detection methods based on neural networks, presenting some promising outcomes that aid in distinguishing between different collision events within the HRC context. Also building from this point, Sharkawy et al. [14] probe deep into collided link identification through neural network applications for collision detections, thereby stressing the need for accurate detection methods that are always responsive. Further research by Sharkawy, Koustoumpardis, and Aspragathos [15] investigate specific manipulator designs that use joint position sensors only, when it comes to neural network collision detection while still producing good results. This method represents a move towards cheaper but effective collision detection systems' implementations that can be directly incorporated into existing robotic frameworks. Sharkawy and Mostfa [16] have addressed the issue of training and developing neural systems that ensure safe human-robot collaboration; their methods are meant to improve learning in such networks. What they have done underscores the need for strong training protocols to ensure collaborative robots can always work correctly. Heo et al. [17] propose a deep-learning solution for collision detection on industrial cooperative robots, thus contributing towards intelligent and autonomous safety systems within an industrial context as described by them; this research also shows how well deep learning may supplement traditional safety measures, thereby quickening response times while reducing downtime. The integration of deep Reinforcement Learning (RL) to improve safety in collaborative robots is explored by El-Shamouty et al. [18]. This framework uses deep reinforcement learning to encode task and safety specifications that enable robots to learn how to move without colliding in various scenarios. The proposed methodology mitigates hazards by transferring learning from controlled environments in the labs to industrial applications with little need for extensive risk assessments. Deep RL, according to the study's results, has the ability to make HRC much safer by enabling robots that are capable of learning and modifying their actions in relation to changing surroundings and sophisticated duties. Park et al. [19] address the issue of collision detection without using expensive joint torque sensors. This is achieved by using support vector machines (SVM) and convolutional neural networks (CNN) based on motor current measurements to avoid friction models and manual threshold tuning. The CNN method performs very well when trained with a large data set. In contrast, the SVM method is resistant to overfitting and, hence, effective with fewer samples, meaning that they respond differently depending on the type of collision. A six-degree-of-freedom industrial collaborative robot was employed to validate the effectiveness of these techniques as it demonstrated a precise detection of hard and soft collisions in real-time. Zhou et al. [20] propose a deep learning approach to automate object detection from 3D point clouds in small- and medium-sized enterprises (SMEs). Light inconsistency and structural ambiguity are among the challenges addressed by combining 2D and 3D technology to enhance localization and improve detection accuracy. The system can effectively detect and localize targets using 3D point cloud technology and deep learning through PointVoxel-Region Based Convolutional Neural Networks (PV-RCNN), where 2D cameras facilitate the calibration of precise manipulator positioning. According to their results, under variable lighting conditions, it has high accuracy in detecting objects and automatically completes the planned tasks, demonstrating its reliable performance. Although the solution does not target small-size components, it stresses safety and efficiency, conforms to industry standards, and reduces the manufacturing cycle time. In their study, Popov and Klimchik [21] look into using transfer learning to locate collisions during joint work between humans and robots. They discovered that much bigger datasets or revising large parts of software code could be skipped with transfer learning, hence saving time needed before deploying these systems in different robotic applications where collision detection is frequently required. Additionally, Czubenko and Kowalczuk [22] came up with a basic model of artificial intelligence for detecting when two or more cooperative robots might collide; their research suggests that it would not take much effort to design simple yet effective AI models capable of boosting human-robot safety during interaction processes. Shaji [23] provides some insights into what happens during HRI by providing resource materials alongside GitHub implementation notes so that people can get involved easily even if they have never worked on such projects before, aiming at making the development of HRI systems open and transparent through sharing information widely.

This work investigates how effective various ML and DL models are in recognizing collisions and other types of contact between human operators and manipulator robotic arms. A dataset containing different contact states is used to evaluate three different models, namely, KNN, Bagging, and LSTM networks. The intention is to compare what every method does best or worst so that these reflections may guide us when selecting appropriate models for real-world HRC systems. The classic machine learning algorithm KNN and ensemble learning technique Bagging are taken for practicality and performance assessment. At the same time, LSTM, a kind of recurrent neural network, is considered due to its capability to handle temporal dependencies critical during dynamic human-robot interactions. Recognition and classification of contact states in human-robot collaboration (HRC) relies heavily on machine learning and deep learning models. Such models make distinguishing between complicated interaction contexts possible, which boosts safety and productivity in collaborative settings. The significance of this study lies in its contribution towards bridging the gap between theoretical research and their applications into practice, hence promoting safer, faster, and more intelligent robots.

This article can be divided into several sections. The second section gives detailed information about the dataset used in this research, such as its collection process and a brief explanation of what it contains. This part of the paper also introduces the proposed model, how data is preprocessed, and trained models. Section III demonstrates various performance measurements obtained from experiments done on the models with comparisons and discussions. Section IV concludes this paper and offers an in-depth analysis of the results, which ends with suggestions for future work and a reflection on what could be done to overcome limitations.

II. MATERIAL AND METHODOLOGY

Several machine learning and deep learning models are developed for contact detection in human-robot collaboration to ensure safety. These models include KNN, Bagging, and LSTM as they are proved to be most suitable for the used dataset. The first two models were already tested by Shaji [23], nevertheless, in this work the effect of crucial dataset processing steps is studied and thereafter compared with the powerful performance of LSTM neural network model.

A. Dataset Description

For machine learning models to be effective, they must have a variety of representative datasets. The dataset used in this study provides an essential source for accurate contact state detection using different sensors as input features to the machine learning algorithm. The contact detection dataset that is used in this work is publicly available at [24]. It consists of five contact states: non-contact, where zero contact occurred between the robot and the human; intentional link5 and intentional link6, where a human deliberately grasped the arm for collaborative purposes; and collision link5 and collision link6, where the contact made is incidental and might result in injuries. The reason behind choosing only the last two links of the robot to examine contact situations is the fact that these two are farther from the base of the robot and closer to the human in its workspace, making them more likely the ones experiencing intentional contact or a collision. These five classes represent a comprehensive analogy of collaborative environments, making this dataset a strong candidate to build reliable models that enhance safety in real-life collaboration by accurately detecting collisions.

The Franka Emika Panda cobot was used to collect the dataset, which comprises 2205 samples distributed among the five classes explained earlier; the distribution of classes is depicted in Figure 2.

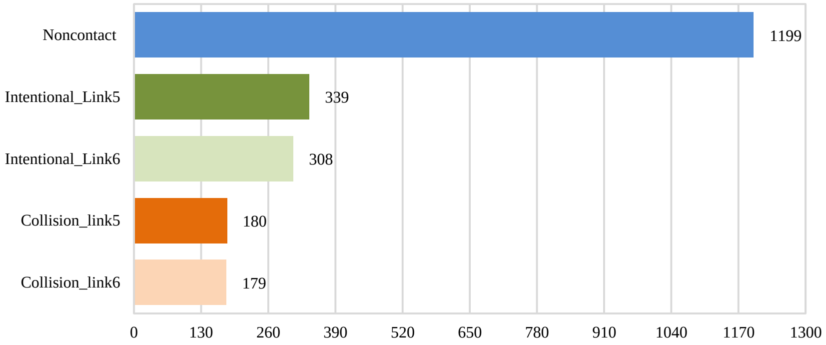

Figure 2. Distribution of the classes in the dataset

Each of these samples represents 140 ms of four sensor data from each of the seven joints of the robotic arm, including the joint torque sensor, the external torque sensor, the joint position sensor, and the joint velocity sensor. A 200 Hz sampling frequency was chosen, which resulted in 28 samples per time lapse [25]. To conclude, the diversity in the contact scenarios along with incorporating several sensors to collect data from, allows for a powerful dataset that aids various models to identify patterns effectively, thereby enhancing their classification performance and achieving safe collaboration.

B. The K-Nearest Neighbors and Bagging Models

KNN and Bagging classifier algorithms are employed to classify contact states. The KNN classifier is specifically beneficial for the used dataset as it leverages the local distribution of data points; it is also simple and effective for handling multi-class classification problems; these qualities make it an effective solution to identify and differentiate intentional contacts and collisions correctly [26]. The bagging technique is also tested because it helps reduce the variance and avoids overfitting; in this model, multiple subsets of the training data are created, and individual models are trained on each subset; the final prediction is made by averaging over the individual models [27]. Shaji Amal [23] originally tested these two models on the same dataset; however, his method involved limited data preprocessing, which could greatly enhance performance. In this work, grid search is used for both KNN and Bagging models in order to optimize parameters through the exploration of numerous combinations. As for the dataset, multiple steps for preprocessing are followed, including normalizing using a StandardScaler, data augmentation by adding noise to the training set, Synthetic Minority Over-sampling Technique (SMOTE) is also applied to handle class imbalance, which is easily noticed in Figure 2, last but not least, the data was shuffled to preserve randomness. For evaluation, performance reports, confusion matrices, and learning curves' plots are used.

C. The Long Short-Term Memory Model

In addition to classical machine learning approaches, deep learning techniques are also explored. Moreover, the LSTM model was chosen because it is the most suitable for detecting temporal dependencies in proprioceptive sensor data and identifying subtle changes and patterns across the time steps in the samples [28]. Data preprocessing for the LSTM model follows a similar procedure as for the KNN and Bagging models: reshaping, normalizing, data augmenting, and computing class weights to apply later at training for balancing the dataset. The architecture of the tested LSTM model consists of two Bidirectional LSTM layers with 128 and 48 units, respectively, followed by Batch Normalization and Dropout layers to prevent overfitting, and concluded with dense layers for classification that utilizes SoftMax activation function for multi-class output. The training process involves using early stopping and learning rate reduction callbacks to avoid overfitting. The Adam optimizer and categorical cross-entropy loss function are also used for compiling.

III. RESULTS AND DISCUSSIONS

This section presents tests and discusses the experimental setup, data splitting, the training of the models with the hyperparameters, and the results obtained using various evaluation techniques.

A. Test Platform

This study was carried out on a personal computer equipped with an Intel Core i7 processor that supports Microsoft Edge Jupyter Notebook. This specification provides enough processing power to handle complex machine learning and deep learning training and evaluation procedures. Python programming language is used along with its numerous packages and modules that aid in creating models, data preprocessing, and visualization. These software tools along with the powerful hardware, result in a flexible and effective environment for training and validating the models.

B. Data Split

Dataset splitting aims to distribute the data to two or three sets for training, testing, and validation. The training portion should have enough amount of data for the model to learn from, and the test portion should also be sufficient for making correct decisions regarding the generalization and performance of the models; the validation data, on the other hand, helps to tune the hyperparameters and in turn avoid overfitting. For the KNN and Bagging models, the dataset is split into 70% training data and 30% for testing. The LSTM model is also trained on 65% of the dataset, with the remaining distributed as 20% for testing and 15% for validation.

C. Training Process and Hyperparameters

The optimum hyperparameters for KNN and Bagging classifiers were obtained from the grid search method and yielded the following for KNN: metric=Manhattan, n_neighbors=3, weights=distance; these choices reveal that the resulting model is suitable for complex data, provides a balance between sensitivity and effectiveness, and provides accurate classification. For the Bagging method, the best parameters are bootstrap=False, bootstrap_features=True, max_features=0.5, max_samples=1.0, and n_estimators=30, these imply that the model is diverse and, in turn better performing, less prone to overfitting, and generalizes well to unseen data.

On the other hand, the LSTM model is designed with the following parameters: BiLSTM layer 1=128 units, BiLSTM layer 2=48 units, both followed by batch normalization, dropout of 0.4, and a dense (fully connected layer). The training process undergoes 100 epochs with 64 batch sizes, and the earlier computed class weights are applied to handle class imbalance. Regularization and callbacks are applied to ensure the model is well-regularized, stable, and effective.

D. Results and Comparisons

The evaluation process, as mentioned earlier, is carried out on the test set. Measures like accuracy, precision, recall, and F1-Score are calculated through the following expressions:

$$Accuracy = (True\ Positive + True\ Negative) / (True\ Positive + True\ Negative + False\ Positive + False\ Negative) \quad (1)$$

$$Precision = True\ Positive / (True\ Positive + False\ Positive) \quad (2)$$

$$Recall = True\ Positive / (True\ Positive + False\ Negative) \quad (3)$$

$$F1\text{-}Score = 2 * (Precision * Recall) / (Precision + Recall) \quad (4)$$

The KNN model achieved an accuracy of 93.9%, and the Bagging classifier achieved a higher accuracy of 95.1%. However, the LSTM neural network had an astonishing score of 98.1%, which outperformed the classical machine learning models. The models generally showcase high rates throughout all metrics, indicating their worthiness in contact state classification. This is evident in the following Tables 1, 2, and 3, and Figures 3, 4, and 5, which illustrate the classification reports plus the confusion matrices, respectively, for each model:

TABLE 1. CLASSIFICATION REPORT FOR THE K-NEAREST NEIGHBORS MODEL

Classes	**Precision**	**Recall**	**F1_Score**
Collision_Link5	0.84	0.89	0.86
Collision_Link6	0.95	0.95	0.95
Intentional_Link5	0.90	0.83	0.86
Intentional_Link6	0.92	0.89	0.90
Non-contact	0.97	1.00	0.99
Macro avg	0.92	0.91	0.91
Weighted avg	0.94	0.94	0.94
Accuracy		0.9396	

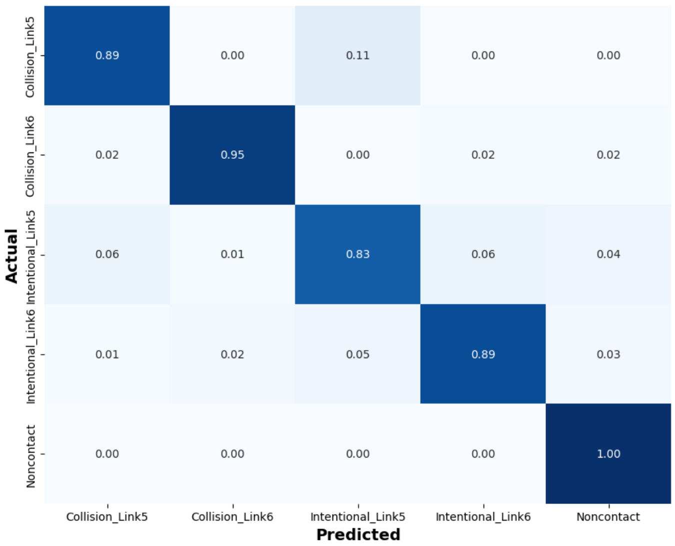

Figure 3. The confusion matrix for the k-nearest neighbors' model

TABLE 2. CLASSIFICATION REPORT FOR THE BAGGING CLASSIFIER

Classes	Precision	Recall	F1_Score
Collision_Link5	0.83	0.92	0.88
Collision_Link6	0.98	0.97	0.97
Intentional_Link5	0.91	0.86	0.88
Intentional_Link6	0.93	0.90	0.91
Non-contact	0.99	1.00	0.99
Macro avg	0.93	0.93	0.93
Weighted avg	0.95	0.95	0.95
Accuracy		0.9517	

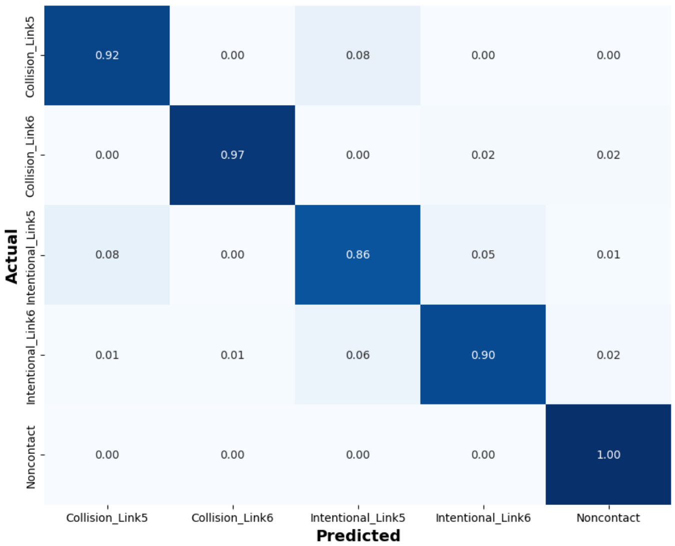

Figure 4. The confusion matrix for the bagging classifier

TABLE **3**. CLASSIFICATION REPORT FOR THE LONG SHORT-TERM MEMORY MODEL

Classes	**Precision**	**Recall**	**F1_Score**
Collision_Link5	0.94	0.95	0.95
Collision_Link6	0.97	0.97	0.97
Intentional_Link5	0.96	0.97	0.97
Intentional_Link6	0.98	0.96	0.97
Non-contact	1.00	1.00	1.00
Macro avg	0.97	0.97	0.97
Weighted avg	0.98	0.98	0.98
Accuracy		0.9819	

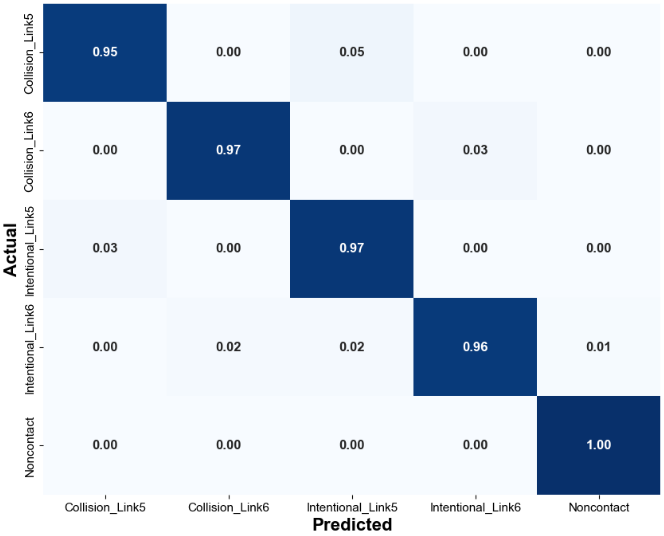

Figure 5. The confusion matrix for the long short-term memory model

As depicted in the above tables and figures, all models perform well, but LSTM proves to be the best among them, and this indicates that there is a need for including advanced neural architectures for high-dimensional and sequential data; it has an improved ability to find out complex patterns more than traditional machine learning algorithms like KNN, or ensemble techniques such as Bagging.

The following Figures 6, 7, and 8 depict the graphs of learning curves for further evaluation of the performance. For the KNN model, the training score remains at perfect accuracy, indicating that the model fits the data very well. Whereas the cross-validation score starts low and then increases with the increase of training size, this suggests that with more training data, the performance improves. The gap between the two curves decreases with the increase of the training size, denoting the improving generalization of the model. The Bagging model fits the training data perfectly, as evidenced by the training score, while the validation score increases and the gap between the training and cross-validation scores decreases with bigger training sizes, indicating the reduced overfitting and the superior performance of the Bagging classifier compared with the KNN model. The last figure illustrates the accuracy and loss graphs of the LSTM model; in the initial epochs, the accuracy graph shows a rapid increase while the loss graph shows a rapid decrease, and both graphs show stabilizing performance. The high accuracy and low loss values suggest that the model is performing very well, learning effectively, and minimizing errors; furthermore, no signs of overfitting are shown in all models.

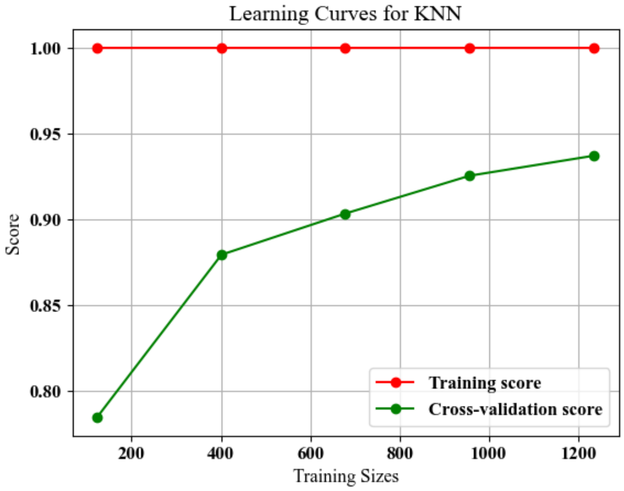

Figure 6. The learning curves for k-nearest neighbors

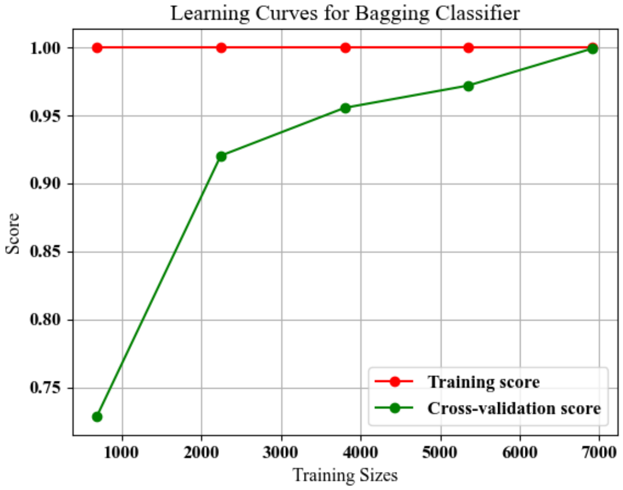

Figure 7. The learning curves for bagging classifier

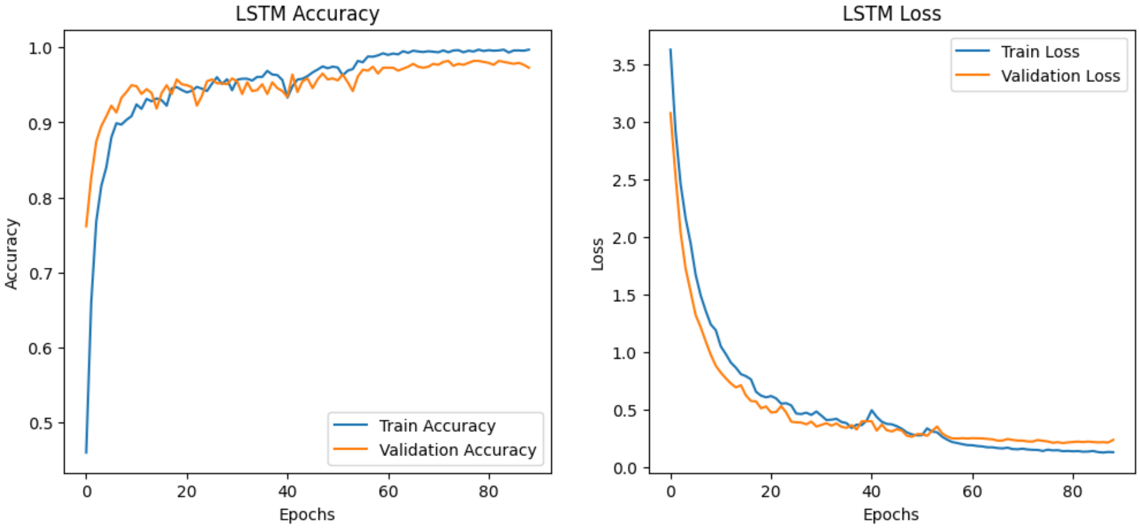

Figure 8. The learning curves for the long short-term memory model

TABLE 4. A PERFORMANCE COMPARISON FOR THE MODELS WITH AND WITHOUT DATA PREPROCESSING

Metric	**KNN**	**Bagging**	**Data Preprocessing**		
			KNN	**Bagging**	**LSTM**
Accuracy	91.6%	92.0%	93.9%	95.1%	98.1%

Table 4 above summarizes a comparison of the accuracy performance of the tested models. In the beginning, KNN and Bagging models' accuracies are presented; these models were tested by Shaji [23]; however, in this work, data preprocessing steps that are found to be necessary for this specific dataset are applied, and a noticeable improvement in the performance against all metrics is observed.

The outperforming results of the LSTM model manifest the significance of deep learning frameworks in HRC systems. Accurately detecting real-time contacts using high-dimensional sensory data confirms the profound impact of machine learning in ensuring safe collaboration.

IV. CONCLUSIONS

To improve safe and efficient human-robot collaboration, this work studies how effective machine learning and deep learning models are at recognizing contact types between humans and robotic manipulator arms. The aim is to design complex systems for detecting contacts that can differentiate non-contact movements, accidental collisions, and deliberate interactions. Focusing on three models: K-nearest neighbors (KNN), Bagging, and Long Short-Term Memory (LSTM) networks. The tests conducted on the contact detection dataset brought about numerous major findings. The KNN model, despite its good performance, could not handle elaborate and high dimensional data well enough and showed a tendency to overfit small datasets.

On the other hand, the ensemble Bagging method helped increase overall performance by reducing variance while making the model generalize better. Nevertheless, the best outcomes were achieved using the LSTM model, which dealt exceptionally well with temporal correlations in time series sensor data from dynamic human-robot interactions. Not only did this LSTM model give higher accuracy rates, but it also proved to be steadier and more robust throughout training as well as test phases than any other model. These findings stress that deep learning models like LSTM are most compatible with the highly dimensional sequential data obtained from the robotic arm. The improved performance of LSTM has shown the necessity of including complex neural network models in creating and operating human-robot collaboration systems.

Furthermore, data preprocessing and hyperparameter optimization are important when improving a model's performance. In this study, the indispensable role played by machine learning in improving HRC systems is demonstrated where the detection of contact states depends heavily on model choice and preprocessing. In real-life applications, this reflects the potential for greater integration of more advanced ML techniques into current methods with regard to the effectiveness of neural network models like LSTM in managing complicated datasets.

Despite the great results, this study has some shortcomings. The dataset used for training and evaluation is small and not diverse enough; therefore, the models could fail to apply to other HRC scenarios. In addition, the tests are performed under controlled environments that do not reflect real-life situations with all their intricacies and uncertainties. To get past these constraints and improve the detection process in HRC, future studies will benefit from increasing the dataset to cover different types of tasks, environments, and contacts and combining multiple sensors to make the models more robust and applicable across different scenarios. Secondly, implementing and testing these models in real-world HRC systems will give better insight into their performance under changing conditions. Last but not least, looking at advanced neural architectures like transformer-based models capable of handling complex temporal and spatial data.

REFERENCES

[1] L. Gualtieri, E. Rauch and R. Vidoni, "Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review," *Robotics and Computer-Integrated Manufacturing,* vol. 67, p. 101998, 2021.

[2] L. N. Faraj and B. M. Albaker, "Comprehensive Review and a New Conceptual Framework for Collision Detection and Avoidance System in Human-Robot Collaboration Environment," in *International Research Conference on Engineering and Applied Sciences,* Baghdad, 2023.

[3] A. C. Simões, A. Pinto, J. Santos, S. Pinheiro and D. Romero, "Designing human-robot collaboration (HRC) workspaces in industrial settings: A systematic literature review," *Journal of Manufacturing Systems,* pp. 28-43, 2022.

[4] R. R. Galin and R. V. Meshcheryakov, "Human-Robot Interaction Efficiency and Human-Robot Collaboration," in *Robotics: Industry 4.0 Issues {\&} New Intelligent Control Paradigms*, Cham, Springer International Publishing, 2020, pp. 55--63.

[5] A. Zacharaki, I. Kostavelis, A. Gasteratos and I. Dokas, "Safety bounds in human robot interaction: A survey," *Safety Science*, vol. 127, pp. 104-667, 2020.

[6] A. Weiss, A. K. Wortmeier and B. Kubicek, "Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human–Robot Collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335-345, 2021.

[7] J. E. Michaelis, A. Siebert-Evenstone, D. W. Shaffer and B. Mutlu, "Collaborative or Simply Uncaged? Understanding Human-Cobot Interactions in Automation," in *Association for Computing Machinery*, New York, NY, USA, 2020.

[8] A. Castro, F. Silva and V. Santos, "Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics," *Sensors*, vol. 21, no. 12, 2021.

[9] M. Rezayati, G. Zanni, Y. Zaoshi, D. Scaramuzza and H. W. v. d. Venn, "Improving safety in physical human-robot collaboration via deep metric learning," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, Germany, 2022.

[10] R. Neurauter, M. Sereinig, P. Manzl, M. Saveriano and e. al, "RoboCup@Work Team Tyrolics Team Description Paper 2021," 2021.

[11] R. Salehzadeh, J. Gong and N. Jalili, "Purposeful Communication in Human–Robot Collaboration: A Review of Modern Approaches in Manufacturing," *IEEE Access*, vol. 10, pp. 129344-129361, 2022.

[12] A. Sharkawy and P. Koustoumpardis, "Human–Robot Interaction: A Review and Analysis on Variable Admittance Control, Safety, and Perspectives," *Machines*, vol. 10, no. 7, 2022.

[13] A. N. Sharkawy AN, "Human-Robot Collision Detection Based on Neural Networks," *Int. J. Mech. Eng. Robot. Res*, vol. 7, pp. 150-157, 2018.

[14] S. AN, K. PN and A. N, "Manipulator Collision Detection and Collided Link Identification based on Neural Networks," in *Advances in Service and Industrial Robotics*, Cham, Switzerland, 2018.

[15] S. AN, K. PN and A. N, "Neural Network Design for Manipulator Collision Detection Based only on the Joint Position Sensors," *Robotica*, vol. 38, pp. 1737-1755, 2020.

[16] A. N. Sharkawy and A. A. Mostfa, "Neural networks' design and training for safe human–robot cooperation," *J. King Saud Univ.-Eng. Sci*, pp. 1-15, 2021.

[17] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robot*, vol. 4, no. 2, pp. 740-746, 2019.

[18] M. El-Shamouty, X. Wu, S. Yang, M. Albus and M. F. Huber, "Towards Safe Human-Robot Collaboration Using Deep Reinforcement Learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020.

[19] K. M. Park, J. Kim, J. Park and F. C. Park, "Learning-Based Real-Time Detection of Robot Collisions Without Joint Torque Sensors," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 103-110, 2021.

[20] Z. Zhou, L. Li, A. Fürsterling, H. J. Durocher, J. Mouridsen and X. Zhang, "Learning-based object detection and localization for a mobile robot manipulator in SME production," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102229, 2022.

[21] D. Popov and A. Klimchik, "Transfer Learning for Collision Localization in Collaborative Robotics," pp. 1-7, 2020.

[22] M. Czubenko and Z. Kowalczuk, "A Simple Neural Network for Collision Detection of Collaborative Robots," *Sensors*, vol. 21, no. 12, 2021.

[23] A. Shaji, "Human-robot-interaction," GitHub, 2023. [Online]. Available: https://github.com/amalshaji4540/Human-robot-interacton.

[24] M. Rezayati and H. W. van de Venn, *Physical Human-Robot Contact Detection*, Mendeley Data, 2021.

[25] F. M. Amin, M. Rezayati, H. W. v. d. Venn and H. Karimpour, "A Mixed-Perception Approach for Safe Human–Robot Collaboration in Industrial Automation," *Sensors*, vol. 20, no. 21, 2020.

[26] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, p. 123–140, 1996.

[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 19, no. 8, pp. 1735-1780, 1997.