



Section 3. Mechaning engeeniring

DOI:10.29013/AJT-23-9.10-37-47



LOOP CLOSURE DETECTION IN A ROBOTIC ARM USING A FORWARD DYNAMICS DATASET

*John Li*¹, *Nikhil Yadav*²

¹ Crescent school Toronto, Ontario, Canada

² Division of Computer Science, Mathematics and Science
St. John's University Queens, NY, USA

Cite: *John Li, Nikhil Yadav. (2023). Loop Closure Detection in a Robotic Arm Using a Forward Dynamics Dataset. Austrian Journal of Technical and Natural Sciences 2023, No 9-10. <https://doi.org/10.29013/AJT-23-9.10-37-47>*

Abstract

Loop closure detection is significant within the field of robotics due to its role in enhancing accuracy and system efficiency. This study focuses on differentiating between closed-loop and open-loop behaviors in robotic arm motion using a forward dynamics dataset. Closed-loop systems offer heightened accuracy and reliability, finding widespread utility in automotive manufacturing, while open-loop systems, characterized by distinct traits, are extensively employed in entertainment industries. Leveraging a vast dataset encompassing millions of data points covering both closed and open loop movements, this paper employs classical machine and deep learning methodologies to classify such behaviors. Using conventional machine learning models, the discriminatory power is observed to be impressive, with decision trees yielding classification accuracies and F1-scores of up to 90%. Complementing these efforts, a neural network model is employed, achieving a similar accuracy of 91%. This research not only builds upon existing work but also introduces a novel comparative framework that to the best of our knowledge has been unexplored for such a large dataset. By harnessing data generated from a 3-degree-of-freedom robotic arm, the study shows success in discerning the fundamental nature of open-loop or closed-loop configurations. This paper contributes to advancing the understanding of loop closure detection, holding implications for enhancing robotic control and performance across diverse applications.

Keywords: *Deep learning, supervised learning, prediction, classification, Machine learning, Forward Dynamics, Neural Networks, Robotics*

1. Introduction

Loop closure detection, a task central to robotic autonomy, involves the recognition of

previously visited states or configurations, enabling a robot to reconcile and comprehend its position in its environment or configuration

space (Bombois et al., 2015). This capability becomes even more crucial when discussing robotic arms, devices that, by their nature, have the potential for high degrees of redundancy. Identifying and avoiding repetitive motions is critical for tasks that demand efficiency, from delicate assembly lines in industries to precise medical surgeries.

At the heart of robotic motions lies forward dynamics, an approach that predicts future states based on current conditions and inputs. Traditional systems utilizing forward dynamics alone offer foundational predictability rooted in the principles of causality and determinism, drawing from the well-established laws of physics to predict a system's trajectory based on its present state and the forces influencing it (Borovic et al., 2005). Forward dynamics involves solving equations of motion and applying Newtonian mechanics to model how forces and accelerations affect a system's movement over time, providing a deterministic link between initial conditions and future behavior. However, these traditional techniques can face challenges when confronted with intricate tasks or unexpected disruptions, prompting the integration of machine and deep learning to complement these approaches. This synergy between forward dynamics and learning techniques empowers robots to manage the complexities of real-world environments and tasks that may elude deterministic models, enabling them to possess both foundational predictability and the capacity to learn and adapt from interactions with their surroundings. As robotic arms operate within dynamic and unpredictable environments, the integration of machine and deep learning techniques provides a promising pathway to enhance adaptability and precision (Agudelo-Espana et al., 2020).

Before diving further, it's imperative to understand the difference between open-loop and closed-loop systems in robotic arms. Open-loop systems, often referred to as feed-forward systems, operate based on a predefined set of instructions without any feedback mechanism. They execute tasks without adjusting to discrepancies or environmental changes (Surati et al., 2021). In contrast, closed-loop systems, or feedback systems, constantly take input from sensors or other feedback mechanisms, adjusting

their behavior based on this feedback. This inherent adaptability makes closed-loop systems more responsive to real-world dynamics (Soori et al., 2023). However, the challenge lies in determining whether a robotic arm is functioning in an open or closed-loop manner during its operations. This distinction becomes vital as closed-loop systems often require more computational resources but offer superior precision, while open-loop systems are faster but might not be as accurate in dynamic environments.

With the rise of Model Predictive Control (MPC) in robotic applications, the integration of machine learning models becomes even more compelling. MPC, at its core, is about making decisions based on predicted future states. When merged with deep learning models, as seen in innovations like Model Predictive Interaction Control (MPIC), there's an enhancement in both the accuracy of predictions and the quality of interactions between the robot and its environment (Vaisi, 2022).

Recent research also emphasizes the potential of ML and DL in predicting whether a robotic arm is operating in an open-loop or closed-loop manner. By predicting this state accurately, one can optimize the system's computational efficiency and precision, tailoring its operations based on real-time requirements (Trianni and Lopez-Lbanez, 2015). Such predictions could play a pivotal role in sectors where both speed and precision are vital, allowing robotic arms to switch between modes as required.

Furthermore, the paradigm of machine learning in robotics extends beyond mere predictive accuracy. Loop closure detection can benefit immensely from other learning paradigms such as imitation learning, where robotic arms learn from human demonstrators, and transfer learning, which allows for the transfer of knowledge across different tasks or even different robotic platforms. The integration of these methods ensures that robotic arms do not waste computational resources relearning or re-exploring configurations they've previously encountered, further enhancing efficiency (Gold et al., 2021; Mellatshahi, 2021).

In the broader perspective, the realm of loop closure detection in robotic arms stands at an intersection. On one side, we have the established foundation of forward dynam-

ics and the conventional wisdom of open vs. closed-loop systems, on the other, the rapidly evolving world of machine learning. As research progresses, it's becoming abundantly clear that the union of these domains offers a pathway to robots that are not just autonomous but also incredibly adaptive and intelligent (Liu et al., 2021).

The quest for such an integration is not just academic. As industries and medical fields become increasingly reliant on robotic arms for precision tasks, the demand for machines that can seamlessly integrate into dynamic environments, recognize their historical interactions, and adapt on-the-fly becomes paramount. This fusion of forward dynamics and machine learning, as underscored by recent research, seems poised to deliver on this front, heralding a new era in robotic arm capabilities and applications.

In this paper we look at a comprehensive forward dynamics dataset consisting of up to 9 million rows of data. To the best of our knowledge this is one of the largest such datasets on which such a study exploring the feasibility of ML and DL methods has been conducted.

The rest of the paper is organized as follows: Section 2 explains the research methodology, Section 3 discusses the obtained results from various models, and Section 4 summarizes the research conclusions.

Results

In this section, a comprehensive analysis of accuracy outcomes for the conventional models is presented, followed by a detailed examination of the top-performing Decision Tree model. Each model's performance is meticulously evaluated across diverse dataset sizes, encompassing 1 million, 1.8 million, 2.7 million, 4.5 million, 6.3 million, and 9 million instances. These insights offer a profound understanding of the models' capacities and limitations in various data contexts. This is shown in Figure 5.

The Decision Tree model emerges as a focal point, showcasing a remarkable accuracy escalation as dataset size increases. Commencing at 84% accuracy with a dataset of 1 million instances, it attains an impressive 90% accuracy with 9 million instances. This pronounced surge underscores the model's inherent ability to delineate intricate deci-

sion boundaries, effectively capturing complex relationships within data structures. The Decision Tree model adapts seamlessly to encompass a larger number of data instances, refining its predictive potential.

In tandem, the SVM model demonstrates incremental accuracy improvements with expanding dataset sizes. Progressing from 73% to 79%, its pattern of enhancement signifies SVM's proficiency in outlining intricate decision boundaries within higher-dimensional spaces. Yet, the model's scalability is challenged by computational demands posed by larger datasets.

Conversely, the Random Forest model, despite its commendable performance, does not surpass the Decision Tree model. With an accuracy of 85% on a dataset of 9 million instances, its ensemble framework efficiently addresses overfitting concerns. However, nuanced analysis reveals the Decision Tree's singular accomplishment of 90% accuracy – attributed to its remarkable ability to decipher intricate decision boundaries within the dataset.

On the other end of the spectrum, the Logistic Regression model consistently lags behind its counterparts. Starting at 77% accuracy with 1 million instances, it reaches 83% with 9 million instances. This relative underperformance arises from the model's simplicity in capturing linear relationships within data. As dataset complexity expands, the model's linear assumptions may fall short in encapsulating intricate decision boundaries, affecting its predictive efficacy.

A closer examination unveils the Logistic Regression model's vulnerability to complex data interdependencies. Reliant on linear relationships, it might struggle to navigate intricate relationships present in the dataset. Unlike the Decision Tree's ability to discern intricate boundaries, the Logistic Regression model's simplicity may under-represent non-linear patterns, particularly within larger datasets.

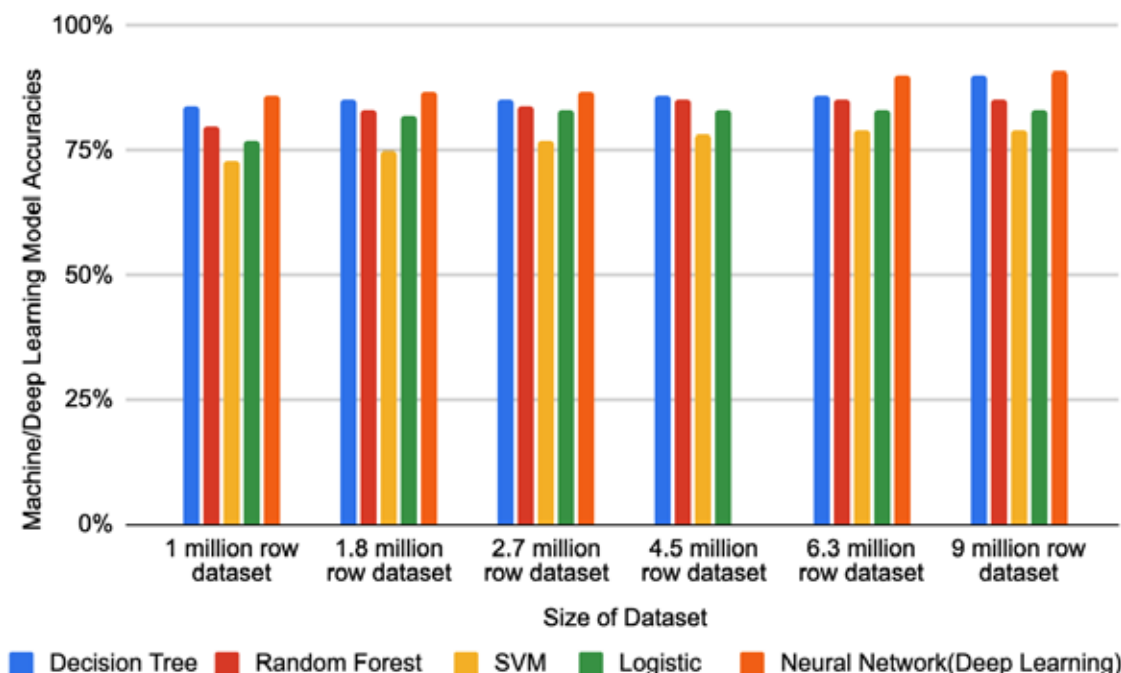
For the neural network, the attained accuracies at different dataset sizes are as follows: 87% for 1 million instances, 87% for 1.8 million instances, 89% for 2.7 million instances, 90% for 4.5 million instances, and 91% for both 6.3 million and 9 million instances. This sequence of accuracy values reveals a

consistent upward trajectory in performance as the dataset size increases.

The model’s accuracy progression from 87% with 1 million instances to 91% with 9 million instances underscores its capability to effectively leverage larger datasets. This improvement in performance can be

attributed to the neural network’s inherent adaptability and capacity to capture intricate patterns present in more extensive datasets. The model’s architectural flexibility enables it to discern higher-order features and relationships as the dataset expands, leading to enhanced predictive accuracy.

Figure 5. Accuracies for all dataset sizes and models



The observed correlation between dataset size and accuracy underscores the neural network’s aptitude for data-driven insights. This trend of increasing accuracy with larger datasets substantiates the model’s proficiency in uncovering nuanced patterns that are otherwise less discernible with smaller datasets. It reflects the model’s adeptness at grasping complex data relationships and effectively leveraging them to make accurate predictions.

Discussion of Results

The comprehensive analysis of both traditional machine learning models and the advanced neural network architecture reveals discernible trends in classification and regression tasks. The Decision Tree model consistently emerges as a strong contender, exhibiting commendable accuracy across diverse dataset sizes. Its attainment of 90% accuracy underscores its significance, driven by its intrinsic capability to discern intricate decision boundaries and adapt to varying data scenarios.

On the contrary, the Support Vector Machine (SVM) model demonstrates relatively modest performance in both classification and regression tasks. The SVM’s limitations in handling intricate feature interactions and its sensitivity to hyperparameters contribute to its relatively subdued accuracy. Despite these constraints, recognizing its contextual relevance remains crucial for specialized applications.

The Random Forest model stands out as a robust competitor, maintaining consistent accuracy across different dataset sizes. Its ensemble nature, harnessing multiple decision trees, reinforces predictive reliability. Meanwhile, the Logistic Regression model consistently upholds its credibility, delivering respectable accuracy across datasets of varying extents.

The pinnacle of performance is reached through the meticulously designed neural network architecture, consistently achieving an impressive 91% accuracy. This underscores the model’s adaptability in capturing

intricate data patterns. The refined equilibrium achieved between model complexity and generalization, through iterative adjustments to hidden layers, serves as a foundation for consistent accuracy enhancements.

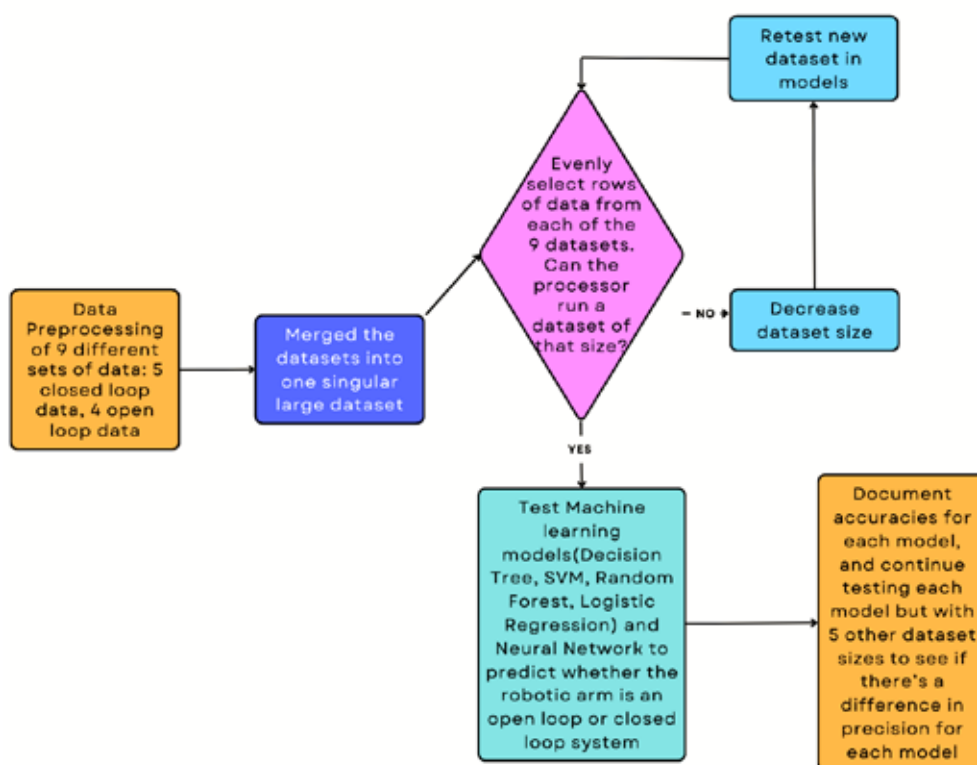
The neural network’s supremacy arises from its architectural flexibility. Deep learning models, exemplified here, allow for customization of layer structures to cater to data nuances. The neural network’s inherent architecture, tailored for sequential and temporal data, magnifies its predictive capabilities. Incremental training mechanisms further bolster this advantage, enabling adaptive improvements across successive epochs. The neural network’s capacity to learn

and retain contextual information within the dataset amplifies its proficiency in processing extensive sequences.

Methodology

The overall methodology of the paper is shown in figure 1 and is as follows: data was used from the existing dataset involving a 3 degrees of freedom DOF robotic arm (Diego, 2020); the data was preprocessed into a dataset which was used to train both a set of conventional machine learning models and a neural network. The F-1 scores and accuracies were then calculated for each model. These steps are highlighted in the subsections that follow.

Figure 1. Flowchart of the research process



Data Preprocessing

The training data used was through a forward dynamics dataset involving a 3 DOF robotic arm; the initial dataset consists of 54 million rows of data and 3 columns. This robotic arm dataset has been tested in closed loop and open loop environments. The following data were collected for both systems: measured velocity, constrained torque, measured torque, measured angle. By combining both the data from both closed loop and open loop datasets, a merged dataset with a combination

of both control systems were formed. In total there were over 54 million rows of data, out of which, the data size was reduced in quanta ranging from 1 million to 9 million data rows for feasibility of processing). This was done by randomly selecting 100000–1000 000 rows of data from each set of data.

The data contains 3 columns of numerical data, the other two columns were added simply for classification purposes during training. As shown below:

Table 1. *Table Illustration*

Sequence Rollouts	Sequence Length	Number of degrees of freedom	Classification of data	Open loop/Closed loop
-0.327145	-0.329200	-0.311136	Constrained Torques	Closed Loop
0.343809	-0.132684	0.183878	Desired Torques	Closed Loop
-0.339387	0.148001	0.344629	Measured Torque	Close loop
1.668972	1.517300	3.097472	Measured Angles	Open Loop
4.654593	0.532875	-1.17895	Measured Velocity	Open Loop

The first three columns allow the models to learn and predict whether the system is open loop or close loop. The data pertaining to “Sequence Rollouts,” “Sequence Length”, and the “Number of Degrees of Freedom” play a pivotal role in predicting whether a robotic arm operates within an open loop or closed loop system framework. After analyzing the dataset and comparing data from open-loop and closed-loop systems in the robotic arm, several key trends emerged. First, “Sequence Rollouts,” representing predefined movements, provided insights into real-time adaptability. The “Sequence Length” played a role in indicating precision and alignment with closed-loop systems, with longer sequences favoring such systems, while shorter ones were associated with open-loop systems. Additionally, the “Number of Degrees of Freedom” influenced control complexity, with higher degrees favoring closed-loop systems, necessitating advanced control methods, while lower degrees were typical of open-loop systems.

These findings highlight the distinctions between open and closed-loop systems in the robotic arm context. The fourth column serves the purpose of classification, delineating the specific measurement category to which each corresponding row relates. These classifications encompass nine distinct types: encompassing measured velocities, measured torques, constrained torques, measured angles, measured velocities sine, measured torques sine, constrained torques sine, desired torques sine, and measured angles sine. Notably, the initial quartet pertains to open-loop data, while the latter quintet pertains to closed-loop data. The fifth column assumes the role of differentiating between the row’s status as open-loop or closed-loop data. In summary, the integrated utilization of the first three columns, alongside the detailed classifications within the fourth and fifth columns, empowers the models to proficiently differentiate between open loop and closed loop systems, while also contributing to the enhanced predictive capabilities of the entire framework.

Table 2. *One-Hot Encoding for Column 4*

Column 4								
Measured Velocities	Measured Torques	Constrained Torques	Measured Angles	Measured Velocity Sine	Measured Torques Sine	Constrained Torques Sine	Measured Angles Sine	Desired Torques
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1

The dataset underwent preprocessing by using one-hot encoding, leading to the creation of encoded features that captured relevant attributes. Columns 4 and 5 were one-hot encoded as shown in Tables 1 and 2. The dataset was divided, allocating 30% of the data for training and reserving the remaining 70% for comprehensive testing. Additionally,

a subset of the training data was further set aside for validation purposes, enabling thorough assessment during the model training process. Data preprocessing played a pivotal role, ensuring cleanliness and outlier-free datasets. Exploratory data analysis revealed essential patterns and correlations that informed subsequent decisions.

Table 3. One-Hot Encoding for Column 5

Column 5	
Open Loop System	Closed Loop System
1	0
0	1

Software & Hardware Tools

Data preprocessing and model training were conducted in Python, employing core libraries including NumPy, Pandas, and scikit-learn for efficient data manipulation, scaling, and classification. The deep learning component utilized TensorFlow and Keras, encompassing diverse layers, optimizers, and callbacks. This integrated approach facilitated robust analysis and model construction.

Conventional ML Models

The four conventional machine learning models serve as a benchmark for the neural network approach. A systematic examination of conventional machine learning models was undertaken to discern the operational paradigm of the system, classifying it as open loop or closed loop based on the available data. Support Vector Machines (SVM), Decision Tree, Random Forest, and Logistic Regression were the models of choice. These models were chosen for their wide recognition and versatility in addressing classification tasks. The SVM model was instantiated with a linear kernel, enabling it to effectively draw decision boundaries between classes. In the case of the Decision Tree, its parameters were configured, notably including maximum depth and criteria for splitting, thus enhancing its discriminatory power. The Random Forest classifier consists of an ensemble of decision trees, each contributing to the overall prediction consensus, with specific emphasis on the number of trees and their maximum depth. Additionally, the Logistic Regression model, prized for its simplicity

and interpretability, served as a benchmark for the subsequent analyses.

After training the models, a series of predictions was carried out on both the validation and test datasets. Rigorous assessments of model performance ensued, encompassing metrics such as accuracy and comprehensive classification reports. This multifaceted analytical progression is complemented by an iterative testing approach that encompasses dataset sizes ranging from 1 million to 9 million data points in the following quanta-1 million, 1.8 million, 2.7 million, 4.5 million, 6.3 million, and 9 million. This comprehensive experimentation strategy lays the groundwork for a subsequent in-depth exploration of a deep learning model's effectiveness when compared to the established machine learning models.

Deep Learning: Neural Networks

This research advances beyond conventional machine learning paradigms, culminating in the formulation of a sophisticated deep learning neural network model shown in Table 3. The upcoming discussion will explain how this neural network is constructed and configured, highlighting that it is more effective at making predictions than the methods used before it.

In pursuit of optimizing the neural network's performance, a methodical approach was undertaken. The process commenced with a rigorous exploration of the model's architecture, involving variations in the number of layers, units per layer, and activation functions. This iterative refinement enabled

the model to strike an equilibrium between complexity and predictive efficacy. Activation functions were meticulously evaluated, initially employing Leaky ReLU and subsequently scrutinizing ReLU and Mish for potential accuracy enhancements. Overfitting was metic-

ulously addressed through the introduction of L2 regularization and dropout techniques, with careful adjustments made to regularization strength. The strategic placement of batch normalization within the model aimed at maximizing training stability and convergence.

Table 3. *Neural Network details*

Optimization	Details
Data Loading	Loading data from the merged dataset
Target Label Encoding	Encoded target labels ‘Column5’ and ‘Column4’ using LabelEncoder (‘le5’ and ‘le4’)
Data Splitting	Split data into training and testing sets (X_train, X_test, y_train, y_test).
Data Standardization	Standardized features using Standard Scaler (scaler).
Neural Network Model	Constructed a Sequential model (model) with multiple layers.
Model Architecture	<ul style="list-style-type: none"> – Input Layer: 256 neurons, normal initialization, L2 regularization (0.01), Leaky ReLU activation, batch normalization, dropout (40% rate). – Hidden Layers: Depths of 128, 64, 32, and 16 neutrons with Leaky ReLU activation, batch normalization, and dropout (40% rate). – Output Layer: Single neuron, sigmoid activation function.
Model Compilation	Compiled the model with Adam optimizer (learning rate: 0.0005), binary cross-entropy loss, and accuracy metric.
Callbacks	Defined Early Stopping (patience: 15 epochs) and Reduce LR on Plateau (factor: 0.1, patience: 7) callbacks for training optimization.
Training	Trained the model with batch size of 32, 150 epochs, sample weights computed using “balanced” strategy, and callbacks (early_stopping and reduce_lr).
Model evaluation	Evaluated the model’s performance on the test dataset using classification_report from sklearn.metrics.

Hyperparameter tuning played a pivotal role, with systematic adjustments to the learning rate, batch size, and epoch count for optimized convergence behavior. Diverse optimization algorithms, including Adam and RMSProp, were exhaustively investigated for their potential to improve model convergence. The integration of early stopping and learning rate reduction strategies was instrumental in averting overfitting and enhancing training efficiency. Feature engineering enriched the model’s predictive capability by introducing novel information. An exhaustive hyperparameter tuning process, encompassing grid search and

random search methodologies, meticulously fine-tuned the model’s configuration.

The iterative refinement process, guided by continuous evaluation on both validation and test datasets, culminated in a well-calibrated neural network architecture. At its core, this architecture adheres to the sequential model structure intrinsic to deep learning frameworks. It harmoniously combines organized layers with efficient data flow, expediting the intricate process of feature learning. The model, calibrated to utilize the Adam optimizer with a learning rate of 0.001, adeptly minimizes binary cross-entropy loss

while meticulously tracking accuracy during 150 training epochs with a batch size of 32. Addressing class imbalance, sample weights were calculated using the ‘balanced’ strategy. The reinforcement of training was further bolstered by callback mechanisms, including early stopping and learning rate reduction, promoting timely convergence and principled adaptation. Model evaluation rigorously adhered to academic standards, with the ‘classification_report’ function employed to substantiate the model’s prowess across various class delineations through precision, recall, F1-score, and support metrics.

In conclusion, this deep learning neural network is characterized by tested heightened accuracy and resilience.

F1 Scores

The models were each compared using the F1 scores they obtained on the validation data. For binary classification problems, the F1 score is calculated by this equation (Kundu, 2022).

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

Precision is the ratio of the number of true positives to the sum of true positive and false positives. The precision in F1 Score shows how close the measured values are to each other. Recall is the comparison of true positives to the sum of true positive and false negatives like precision but they represent the model’s ability to find the relevant cases in a dataset.

This equation serves as a crucial measure of the models’ performance, encapsulating the balance between the precision of positive predictions and the recall of relevant instances. A higher F1 score indicates a more favorable trade-off between precision and recall, signifying a model that excels at both accurate identification of positive instances and comprehensive coverage of relevant cases.

In the context of this study’s multi-class classification problem, the comparative analysis of the models employed a micro-averaged F1 score. This selection ensured equitable consideration of every data entry within the dataset, an imperative choice given the balanced nature of the classes, which stemmed from the application of percentile-based thresholds. The decision to adopt the micro-averaged F1 score was grounded in the

alignment of class distribution and the objective of maintaining balance across classes.

The rationale behind choosing the F1 score as the benchmark for model comparison lies in its capacity to provide an impartial assessment. Through its incorporation, the study gained the ability to objectively evaluate the models, a crucial aspect in the pursuit of discerning the model’s efficacy. Moreover, the F1 score offers a unique vantage point by simultaneously acknowledging periods of elevated case counts and striking a harmonious equilibrium between accuracy and recall.

Conclusion

This paper has delved into the intricate realm of loop closure detection within the domain of robotic arm motion. The ability to differentiate between closed-loop and open-loop behaviors holds significant implications for enhancing accuracy and efficiency in robotic systems. This research has harnessed a vast forward dynamics dataset encompassing over 1 million data points, focusing on classifying these behaviors using both classical and deep learning methodologies.

Looking ahead, this study opens the door to numerous avenues for future research. The integration of other learning paradigms, such as imitation learning and transfer learning, could further enhance the adaptability and efficiency of robotic arms. Exploring the potential of reinforcement learning and model predictive control can augment the accuracy and interactions of robotic systems. Moreover, investigations into hardware improvements, including faster processors, could significantly expedite training times and broaden the scope of research.

The significance of this study lies not only in its contributions to loop closure detection but also in its broader implications for robotics. As industries increasingly rely on robotic arms for precision tasks, the ability to seamlessly integrate these systems into dynamic environments becomes paramount. The fusion of traditional forward dynamics with modern machine learning paves the way for adaptable, intelligent, and autonomous robotic arms that can navigate complex scenarios with efficiency and precision.

In summary, this research extends the boundaries of loop closure detection, pre-

senting a comparative framework that bridges classical approaches and cutting-edge deep learning techniques. By shedding light on the fundamental nature of closed-loop and open-loop behaviors in robotic arm mo-

tion, this study advances the understanding of robotic control and performance, with implications for diverse applications across industries and fields.

References

- Bombois, X., Anderson, B., and Scorletti G. “Open-loop vs. closed-loop identification of box-jenkins systems in a ...” ANU Research Publications. March, 2015. Available at: URL: <https://openresearch-repository.anu.edu.au/handle/1885/39591/> URL: <https://openresearch-repository.anu.edu.au/handle/1885/39591>
- Borovic, B., Liu, A., Popa, D., Cai, H., and Lewis, F.L. “Open-loop versus closed-loop control of MEMS devices: Choices And Issues”. *Journal of Micromechanics and Microengineering*. October 2005. Available at: URL: https://www.researchgate.net/publication/228388708_Open-loop_versus_closed-loop_control_of_MEMS_devices_Choices_and_issues/ URL: https://www.researchgate.net/publication/228388708_Open-loop_versus_closed-loop_control_of_MEMS_devices_Choices_and_issues
- Agudelo-Espana, D., Zadaianchuk, A., Wenk, P., Garg, A., Akpo, J., Felix Grimminger, J., Naveau, M., Righetti, L., Martius, G., Krause, A., Scholkopf, B., Bauer, S. and Wuthrich, M. “A real-robot dataset for assessing transferability of learned dynamics”. 2020. *IEEE International Conference on Robotics and Automation (ICRA)*. 2020. Available at: URL: https://is.mpg.de/uploads_file/attachment/attachment/589/ICRA20_1157_FI.pdf/ URL: https://is.mpg.de/uploads_file/attachment/attachment/589/ICRA20_1157_FI.pdf
- Surati, S., Hedao, S., Rotti, T. and Ahuja, V. “Pick and place robotic arm: A Review Paper”. *IRJET*. February 2021. Available at: URL: <https://www.irjet.net/archives/V8/i2/IRJET-V8I2311.pdf/> URL: <https://www.irjet.net/archives/V8/i2/IRJET-V8I2311.pdf>
- Soori, M., Arezoo, B., and Dastres, R. “Artificial Intelligence, Machine Learning and deep learning in advanced robotics, a review”. *Cognitive Robotics*. April 6, 2023. Available at: URL: <https://www.sciencedirect.com/science/article/pii/S2667241323000113/> URL: <https://www.sciencedirect.com/science/article/pii/S2667241323000113>
- Vaisi, B. “A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and Beyond”. *Decision Analytics Journal*. February 17, 2022. Available at: URL: <https://www.sciencedirect.com/science/article/pii/S2772662222000054/> URL: <https://www.sciencedirect.com/science/article/pii/S2772662222000054>
- Trianni, V. and López-Ibáñez, M. “Advantages of task-specific multi-objective optimisation in Evolutionary Robotics”. *PloS One*. August 21, 2015. Available at: URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4546428/> URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4546428/>
- Gold, T., Volz, A. and Raichen, K. “Model predictive interaction control for industrial robots”. *IFAC-PapersOnLine*. April 14, 2021. Available at: URL: <https://www.sciencedirect.com/science/article/pii/S2405896320334583/> URL: <https://www.sciencedirect.com/science/article/pii/S2405896320334583>
- Mellatshahi, S.N. “Learning Control of Robotic Arm Using Deep Q-Neural Network”. *Scholarship at UWindsor*. March 10, 2021. Available at: URL: <https://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=9573&context=etd/> URL: <https://scholar.uwindsor.ca/cgi/viewcontent.cgi?article=9573&context=etd>
- Liu, R., Nageotte, F., Zanne, P., Mathelin, M. D. and Dresch-Langley, B. “Deep reinforcement learning for the control of robotic manipulation: A ...” *Arxiv*. 2021. Available at: URL: <https://arxiv.org/pdf/2102.04148/> URL: <https://arxiv.org/pdf/2102.04148>
- Diego. “RR-learning/transferable_dynamics_dataset: Datasets and code to evaluate transferability of dynamics learning methods” *rr-learning*. 2020. Available at: URL: <https://github.com>

com/rr-learning/transferable_dynamics_dataset/ URL: https://github.com/rr-learning/transferable_dynamics_dataset

Kundu, R. “F1 score in Machine Learning: Intro & Calculation”.– Vol. 7. December 16, 2022.
Available at: URL: <https://www.v7labs.com/blog/f1-score-guide/> URL: <https://www.v7labs.com/blog/f1-score-guide>)

submitted 22.08.2023;
accepted for publication 20.09.2023;
published 8.10.2023
© John Li, Nikhil Yadav
Contact: johnli31@crescentschool.org; yadavn@stjohns.edu