



LSTM-Based Early Anomaly Detection for Industrial Robotic Arm Cybersecurity

Joanna Raj

Abstract

Robotic systems have become an increasingly attractive target for attackers, which leads to major losses and disruptions. Manufacturing has become the most targeted industry, with around 27.7% of observed incidents in 2025 (IBM, 2026), and the global average cost of a data breach was \$4.44 million (IBM, 2025). Companies continue to face vulnerabilities in their systems, causing shutdowns and damaging their reputation, despite investing billions in robotics. These risks are extremely critical as robotic systems operate as cyber-physical systems that integrate software, sensors, and communication networks. This means that cyberattacks are able to directly impact physical operations and safety (Fosch-Villaronga & Mahler, 2021). The goal of this project is to explore how an LSTM neural network can improve early anomaly detection and reduce cybersecurity threats in robotic systems to minimize both direct and indirect losses. To do this, I developed and tested an AI-driven anomaly detection framework that can monitor robotic behaviour and look out for unusual patterns in behaviour, which is crucial for industrial environments where abnormal sensor behaviour or system failures can cause damage in manufacturing processes and lead to significant operational damage (Lu et al., 2022). This project will use the RoAD dataset to evaluate an LSTM-based autoencoder for detecting anomalies in robotic arm behaviour. The model achieved an AUROC of 0.71 on the collision subset, which is slightly above the best unsupervised result reported with the dataset (Mascolini et al., 2023), though low precision shows that reconstruction error carries only a limited anomaly signal. The weight and velocity subsets are labeled anomalous throughout, so they cannot be used to judge detection ability.

Introduction

Robotic systems are commonly used in manufacturing to perform repetitive tasks such as welding, assembling parts, and packaging products, and they also operate nonstop. As they run nonstop, they are also exposed to many cybersecurity risks (Kayan, 2025). More factories have begun adopting connected and automated systems, creating more opportunities for attackers to interfere with how these robots operate. Robots are connected to networks, making them easily monitored and controlled, which can expose them to cyberattacks that alter how they move or perform. Modern robots are considered cyber-physical systems, which means that they combine software, communication networks, and sensors to interact with the physical world, causing them to be more vulnerable to attacks that can impact real-world operations and safety (Fosch-Villaronga & Mahler, 2021). Even a small change in their instructions or data can cause the robots to behave abnormally and unsafely. A successful attack can damage equipment, leading to significant financial loss and production shutdowns, and it can also put human lives at

risk if the robot moves in unusual ways around workers. For industrial environments, unusual activities and failures can disrupt the entire manufacturing process, increasing operational costs and reducing system reliability (Lu et al., 2022). These problems create a need for a robust method to detect anomalies or unusual robot behaviour early, before the risk becomes larger. Researchers have started to look at protecting industrial robots from these types of threats, and one large area of work is anomaly detection. Anomaly detection focuses on finding unusual robotic behaviour or signs of attacks. There are many approaches that look at the network traffic and communication with robots and controllers to find patterns that are unusual, which might be an attack, but there are also others that focus on the robot's physical behaviour. These physical behavior methods analyze behavior in joint angles, speeds, and forces. Recent approaches have relied on time-series sensor data, as industrial robots generate large amounts of data constantly, which can be used to detect unusual patterns and system failures (Lu et al., 2022). Datasets like the Robotic Arm Dataset (RoAD) record time-series data of robots doing tasks and label which parts are normal and which parts are abnormal. These datasets can be used to test different methods like simple statistical models, k-nearest neighbors (KNN), and autoencoders that look at the normal behaviour and mark differences in anomalies. More advanced models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been applied to anomaly detection in multi-sensor industrial systems to capture spatial and temporal patterns in the data (Canizo et al., 2019).

There are still a lot of parts to this problem that existing methods do not fully solve. A lot of existing models still look at each time step separately or mainly at how well they can reconstruct a single signal without learning how a robot's movement changes over time as part of a sequence. This is a weakness, as some cyberattacks do not cause immediate damage but instead slowly shift the robot's behaviours to cause long-term damage. Industrial systems generate high-dimensional and complex data from many sensors, making it hard for traditional models to detect gradually developing anomalies (Zhou et al., 2021). A few surveys on robot cybersecurity also mention that intrusion detection systems are not designed for robots specifically and can miss subtle changes that do not show up specifically in network data. This means that dangerous, unusual behaviors can go unnoticed, and the robot has already damaged equipment, made defects, or created unsafe conditions.

This project addresses this problem by using a Long Short-Term Memory (LSTM) neural network to model how an industrial robotic arm typically behaves over time. LSTMs are a type of recurrent neural network that work well with time-series data since they can remember patterns and relationships that happen over many time steps and not just one. Sequence-based models like RNN encoder-decoder architectures perform well in anomaly detection by comparing expected and observed behavior over time, making them effective at detecting subtle changes in a system's patterns (Park & Yun, 2018).

For this work, the LSTM model is trained on the RoAD dataset's time-series signals, such as joint positions, to learn what normal robot motion looks like during regular tasks, and it flags sequences that look abnormal (Mascolini et al., 2023). This project's goal is to show how a

sequence-aware approach can be more effective than simpler reconstruction-based models, as it can help to detect anomalies earlier and more accurately and help factories catch cyberattacks or abnormalities before any major damage.

Literature Review

Threat Model

Robotic systems that are used for industrial manufacturing are exposed to many cyber-physical attacks that can change physical processes, steal data, or disrupt networks. These systems are called cyber-physical systems, which means they involve sensors, software, and communication networks to interact with the physical world, creating opportunities for system failures and malicious attacks (Fosch-Villaronga & Mahler, 2021). Examples of this are the Stuxnet worm that targeted Siemens PLCs to slowly change speeds while feeding falsified sensor data into operators, showing that carefully made malware can cause great physical damage and be able to remain hidden from traditional monitoring. These attacks show the need for detection methods that look closely at the behaviour of industrial equipment, like how robots and PLC-controlled devices move and operate, instead of just relying on network traffic and perimeter defenses. Attackers can also manipulate communication data and sensor inputs, which can cause unintended or unsafe actions in physical processes (Alruwaili et al., 2025). Cyber-physical attacks can change motion commands, steal information, stop production, or disrupt timing in a way that can destroy products/equipment, making it important to find a way to detect unusual robot behavior to keep factories safe.

Prior Work in Anomaly Detection

Recent research has addressed this type of threat through learning-based anomaly detection for manufacturing robots, focusing on identifying deviations in their motion or operation. The Robotic Arm Dataset (RoAD) consists of time-series recordings of a robotic arm performing tasks with labeled anomalies, such as point and collective deviations from expected trajectories. Other datasets, for example, RAD (Zhou et al., 2024), build on this idea that capturing anomalies using a real robot for tasks that are then visually observed, focusing on the importance of sensor- and perception-level monitoring for strong anomaly detection. Some approaches rely on sensor-based monitoring; abnormal signals can indicate system failures or cyberattacks that can disrupt the entire manufacturing process (Lu et al., 2022). Many models have been applied to these datasets, from classical outlier detection techniques to deep learning-based methods. Advanced work has used CNN-RNN architectures for industrial anomaly detection, which allows systems to capture spatial and temporal patterns in multi-sensor environments (Canizo et al., 2019). RNN encoder-decoder models have also been used to detect anomalies by comparing expected and observed sequences of machine behaviour over time, showing strong results in industrial settings (Park & Yun, 2018).

Limitations of Existing Approaches

This project focuses on behavioral anomaly detection for industrial robotic systems and not network-level security mechanisms or other protocols. It does not address firewall design, intrusion-prevention systems, or more secure communication schemes.

Network-based intrusion detection systems are commonly used, but they may struggle in detecting subtle anomalies in high-dimensional industrial data streams when dealing with complex sensor inputs (Zhou et al., 2021). Reconstruction-based approaches like an LSTM autoencoder face similar difficulty on the RoAD dataset, as its original authors discovered that autoencoder-based detectors performed close to random chance on some of its subsets, although the anomalies can be separated with a supervised method like logistic regression (Mascolini et al., 2023). This project focuses on learning patterns in robot PLC behaviour and detecting anomalies that could indicate malicious manipulation, even when network indicators look normal.

Proposed Approach

Currently, datasets like RoAD were designed for anomaly detection on a robotic arm and have time-series data of joint positions and other signals, and labels for normal and abnormal behaviour, so it's good to work on this. A lot of RoAD studies focus more on traditional methods or reconstruction-based models like k-nearest neighbors or autoencoders, and don't explore deeper into sequence-aware models that learn about how a robot moves over time. Existing approaches aren't strong with high-dimensional and multi-sensor data, making it difficult to capture complex temporal relationships in industrial systems (Zhou et al., 2021). LSTM-based models, however, are used more in security for tasks like network intrusion detection, where they learn normal event sequences and flag unusual ones (Alexandrov, 2024), but have not been tested carefully on manufacturing robot motions like RoAD. Sequence-based models like LSTMs are designed to learn temporal dependencies in time-series data, which makes them better suited for detecting gradual or subtle anomalies that develop over time instead of appearing instantly (Park & Yun, 2018). In this project, an RNN/LSTM model will be used to train directly on RoAD's time-series signals to learn about normal robotic arm behaviour over time and flag anomalies. The goal of this project is to show that a sequence-aware approach can detect anomalies earlier and achieve better accuracy than simpler models, especially in complex industrial environments where anomalies can develop gradually instead of instantly.

Method

Dataset and Data Preparation

In this project, the Robotic Arm Dataset (RoAD) is used. It is a time-series dataset, collected from an industrial robotic arm that functions under normal and abnormal conditions. The dataset includes multiple records of robotic activity, and each recording has sequential sensor measurements across 86 features, such as joint positions, torques, and other controller-related

variables. Timestamps are used to label whether the system's behaviour is normal or anomalous. The dataset is separated into a training subset that consists of nine recordings of normal operation, and four evaluation subsets covering different conditions: collision, weight, velocity, and a non-anomalous control set.

Before training, all the sensor features were normalized with a min-max scaler. The scaler was fit for only the training subset and applied to the other subsets as well, so that none of the evaluation data had an impact on how the features were scaled.

To prepare the data for time-series modeling, the continuous sensor signals were segmented into fixed-length sequences using a sliding window approach. Each sequence represents a short window of behaviour over a time period. For this project, a length of 50 time steps was used, which allows the model to capture temporal patterns across multiple observations instead of just looking at individual data points on their own. An overlapping window was used with a stride of 1 instead of a non-overlapping one, as this provides more training sequences out of the same nine recordings. This dataset is structured in a three-dimensional structure which consists of the number of sequences, sequence length, and number of sensor features. This allows the model to go through sequential data and learn patterns in robotic actions effectively over time. The normal training sequence was split further into a training set (85%) and a held-out validation set (15%) to assist with model calibration. The validation set was only used for early stopping and for setting the anomaly threshold; it was never used to train the model, but it was kept separate from the collision, control, weight, and velocity evaluation data.

Feature Engineering

Before finalizing this model, I checked if all 86 sensor features were useful or if some of them were adding unnecessary noise unrelated to real anomalies. I compared their distribution in the training set against their distribution in the collision evaluation set for each feature using the Kolmogorov-Smirnov two-sample statistic, which looks at all windows regardless of their label. It simply looks for features whose values shift a lot in training conditions and evaluation conditions and doesn't use any anomaly labels at all. There were 20 features with the largest shifts, which were dropped, and the model was trained on the 66 that remained. The main idea behind this was that features that behave differently between training and evaluation are most likely to throw off the model's reconstruction error in ways that don't have anything to do with the real anomalies.

Model Architecture

This project uses a Long Short-Term Memory (LSTM)-based autoencoder model for anomaly detection. The architecture includes an LSTM encoder that compresses each input sequence into a single latent vector, and a separate LSTM decoder that reconstructs the sequence with only that latent vector, without any direct access to the original input. This is important since in the earlier versions of the architecture, the decoder had access to the encoder's hidden state at every time step, letting information about the current input leak into the reconstruction and

making it harder for the model to differentiate normal and abnormal sequences. The final model consists of three stacked LSTM layers with a hidden dimension of 128 and a dropout of 0.2 between layers.

Training Setup

This model first gets trained on normal training data, where it learns what typical robotic movement looks like before it is introduced to anomalous data. Training reduces the Mean Squared Error between the input sequence and its reconstruction, using the Adam optimizer with a learning rate of 0.001 and gradient clipping to keep training stable. Early stopping was implemented to avoid overfitting due to the limited training data. I also fixed a random seed so that the results could be reproduced; this ended up being extremely important as performance varied across runs before the seed was fixed.

Anomaly Detection Approach

The model is tested on data consisting of both normal and abnormal behaviour after training. The model produces a reconstructed output for each input sequence and also calculates the difference between the input and output as the reconstruction error. Each sequence's label only reflects its final time step, so the anomaly score used here is the reconstruction error at just the final time step of the window, instead of just the error averaged across the whole sequence, keeping the score and label lined up at the same time. A higher error at that final step would mean that the model found that moment harder to reconstruct, suggesting a bigger deviation from normal behaviour.

A threshold is set to decide whether a sequence counts as normal or anomalous. This threshold is set using only the held-out validation data, such as the 95th percentile of the validation reconstruction errors. It is calculated before the model sees collision, control, weight, or velocity data, meaning that none of these labels have influence on where the threshold ends up.

Evaluation Metrics

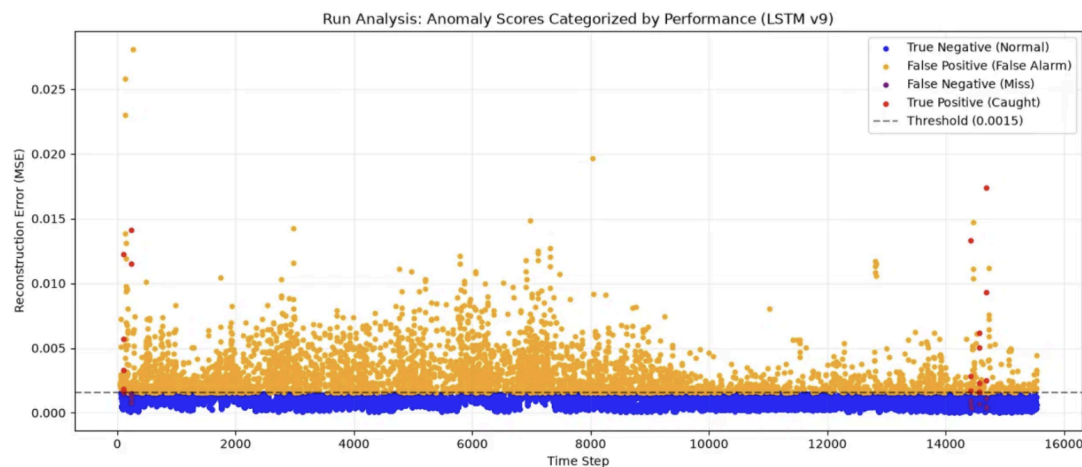
The performance of this model will be evaluated with Area Under the ROC Curve (AUROC) and Area Under the Precision-Recall Curve (AUPRC). These are used since they don't depend on picking one specific threshold and hold up better given how imbalanced the anomaly classes are in this dataset. I also report accuracy, precision, recall, F1-score, and Cohen's Kappa at the chosen threshold with the full confusion matrix (true positives, false positives, true negatives, and false negatives). Reporting the confusion matrix matters since it's possible to get a misleadingly good-looking F1 or accuracy score from a model that's actually just flagging almost everything as anomalous- the confusion matrix makes the results easier to notice. I compared the standard scoring approach- high error would likely mean more anomalous- against an inverted version- lower error most likely anomalous- just to ensure the anomaly signal was pointing in the direction I expected on this dataset.

Results

Overall Performance on the Collision Subset

Using the final model (three-layer LSTM autoencoder with a bottleneck decoder, trained on the 66 selected features with the anomaly threshold set from the held-out validation split), I analyzed performance on the collision subset, containing 49,052 evaluated windows. The threshold was 0.001544, calibrated as the 95th percentile of validation reconstruction errors. The model achieved an accuracy of 72.11%, precision of 0.0312, recall of 0.5840, an F1-score of 0.0593, and a Cohen's Kappa of 0.0316. The confusion matrix is shown below.

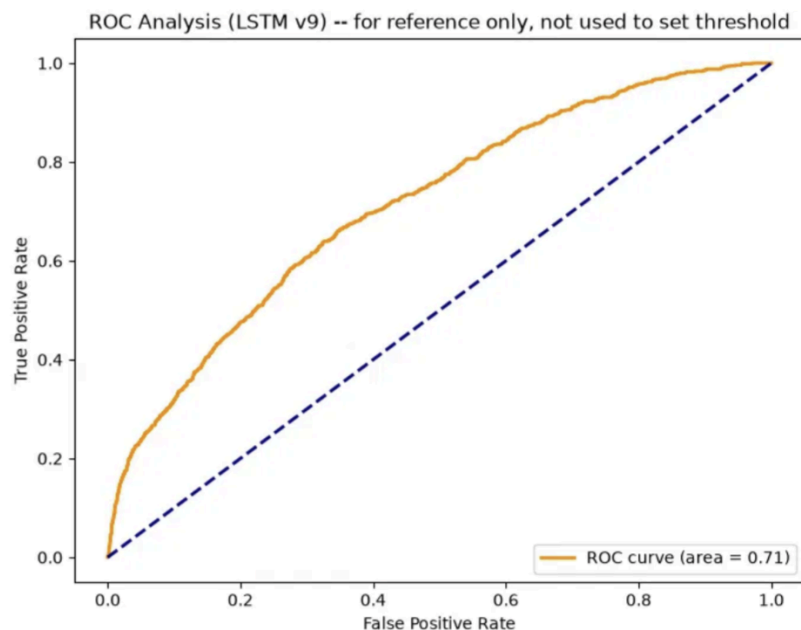
	Predicted Normal	Predicted Anomalous
Actual Normal	34,939 (TN)	13,375 (FP)
Actual Anomalous	307 (FN)	431 (TP)



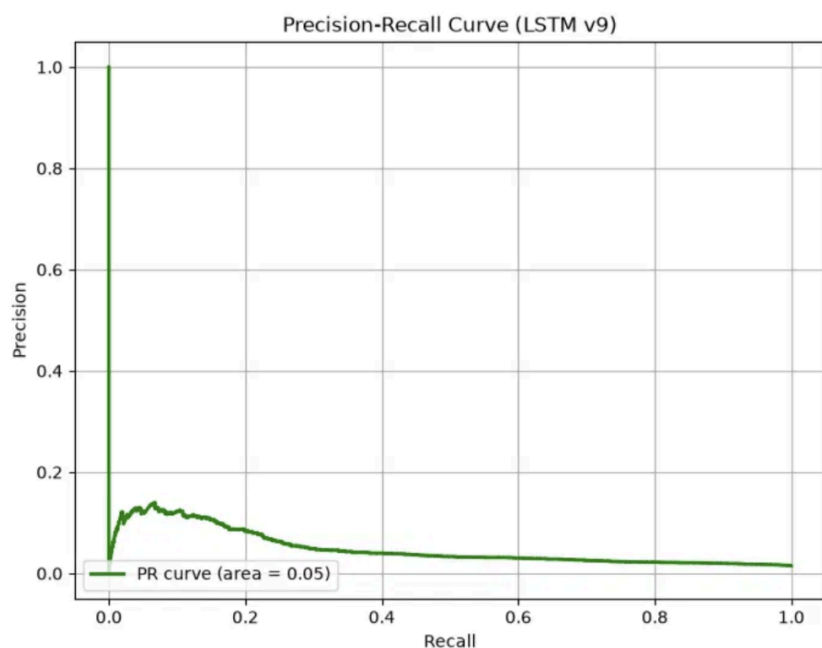
The model was able to correctly identify 431 out of the 738 true anomalies in the collision subset with a recall of 58.4%, while producing 13,375 false alarms. The positive Cohen's Kappa confirms that this level of agreement with the ground-truth labels is better than what would be expected from chance alone, despite the low precision reflecting how rare true anomalies are relative to the overall sample size.

ROC and Precision-Recall Analysis

In order to evaluate the performance independent of any single threshold choice, I computed the ROC curve and AUROC and the Precision-Recall curve and AUPRC.



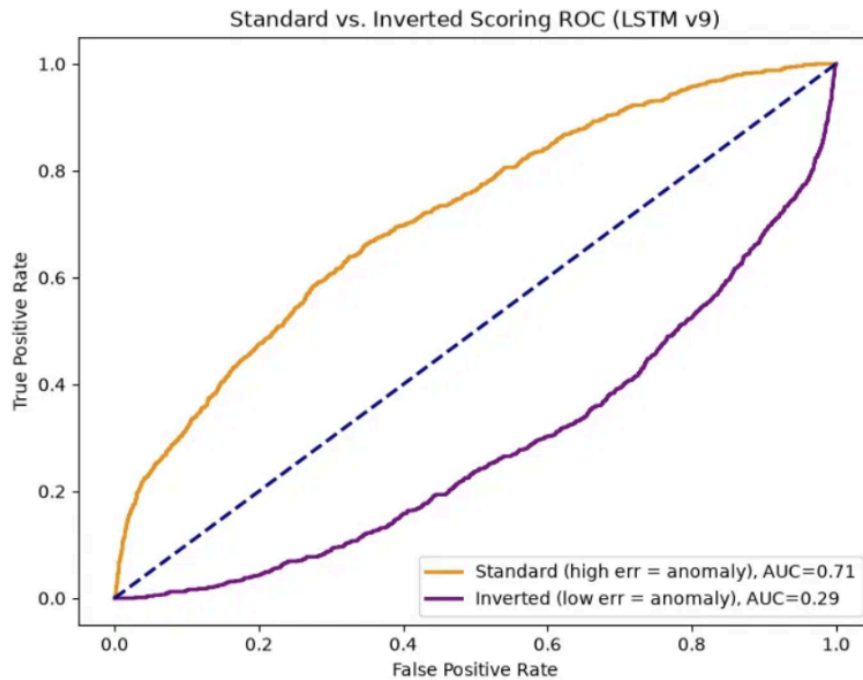
The model had an AUROC of 0.71, meaning the reconstruction-error-based anomaly score could separate normal and anomalous windows above the baseline score of 0.50.



The AUPRC was 0.05. The severe class imbalance in the collision subset means the appropriate random-chance baseline for AUPRC is much lower than 0.5, which is around the proportion of anomalous samples in the dataset, so this value should be interpreted relative to that baseline instead of a fixed threshold of 0.5.

Standard vs. Inverted Scoring

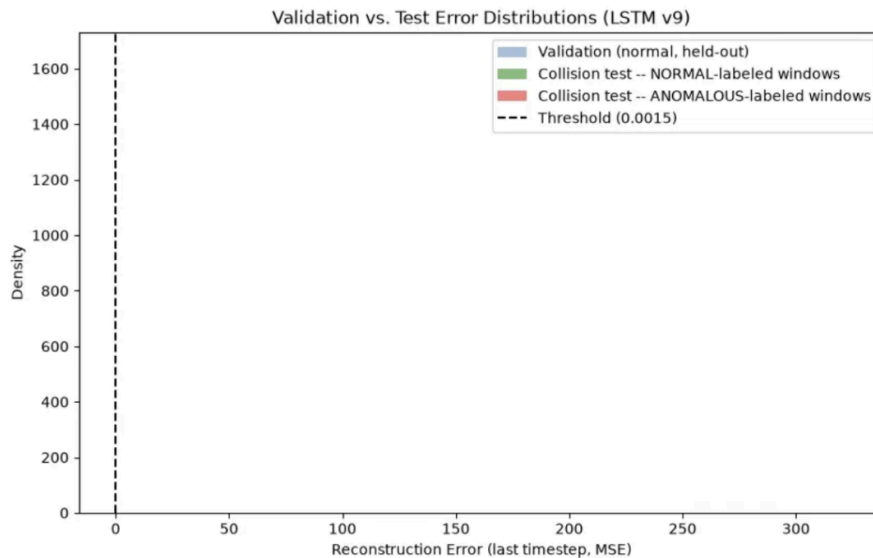
Work that has been done on this dataset before (Mascolini et al., 2023) has raised the possibility that anomalous windows can be reconstructed more accurately than regular windows, which can invert the expected relationship between reconstruction error and anomaly likelihood. To check if this applied to my results, I compared the standard scoring to the inverted scoring.



Standard scoring got an AUROC of 0.71, whereas the inverted scoring got only 0.29, confirming that, for my model, higher reconstruction error is reliably associated with anomalous behaviour, and the standard scoring direction is the correct one to use for this dataset architecture.

Validation vs. Test Error Distribution

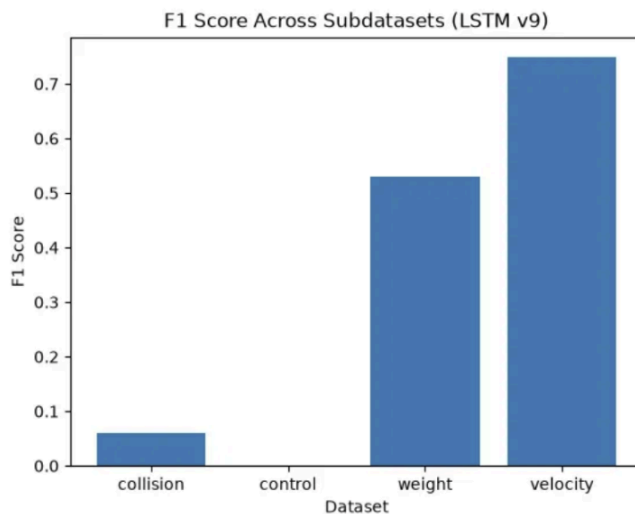
I compared the reconstruction error distribution on the held-out validation set against the error distribution on the collision subset, split by label, in order to check if there are distributional shifts between training and evaluation conditions.



The few extremely outlying windows in the subset with collision led to errors that are much higher than most in the dataset, limiting the visible space on the graph and also making it harder to differentiate the three distribution types at this level of visualization.

Performance Across Subdatasets

Lastly, I evaluated F1-score using the same validation-derived threshold across all four evaluation subsets (collision, control, weight, velocity).



The F1-score on the collision subset was low (0.059) and close to zero on the control subset, but the weight and velocity subsets achieved much higher F1-scores (around 0.53 and 0.75). These two higher values should be read with caution (explained in the Analysis/Discussion below).

Analysis / Discussion

The AUROC of 0.71 and positive Cohen's Kappa (0.0316) demonstrate that the reconstruction error separates normal from anomalous windows better than chance, and the standard-vs-inverted comparison (0.71 vs 0.29) shows that the signal points in the right direction. The low precision (0.0312) and F1-score (0.0593) show how the class imbalance isn't necessarily a failure of the model: a threshold sensitive enough to catch most true anomalies (recall of 58.4%) would also be able to generate many false positives in absolute terms, since anomalous windows are a small fraction of the collision subset.

The F1-score was different on all the subsets - almost zero for collision and control, but higher on weight (0.53) and velocity (0.75). The weight and velocity records are labeled anomalous throughout, however, so these two values show how many windows crossed the threshold, not detection ability. Collision anomalies were also rare (1.5% of windows, so a random AUPRC baseline of around 0.015 when compared to the model's 0.05). The original benchmarks' best unsupervised collision AUROC was 0.68, which is just below the 0.71 here (Mascolini et al., 2023).

Conclusion

This project's goal was to build an anomaly detection system for the Robotic Arm Dataset (RoAD) using an LSTM-based autoencoder. I identified and corrected many methodological issues that had affected results early throughout this project; I fixed mismatches in how the labels and reconstruction scores were calculated, data leakage in threshold selection, and a lack of a true bottleneck in the model's architecture. Doing this was crucial before the model's performance could actually be evaluated properly.

The final model was able to be evaluated with a threshold calibrated from held-out validation data, and it got an AUROC of 0.71 on the collision subset, with a positive Cohen's Kappa and a standard-vs-inverted scoring comparison, confirming the anomaly signal points in the expected direction. These indicate that reconstruction error carries a real, non-trivial anomaly signal on this dataset. The low precision and F1-score at the chosen threshold show a well-known tradeoff in anomaly detection on imbalanced data, rather than a failure to learn anything useful - a high AUROC with a low F1-score is not a contradiction once class imbalance is taken into account.

Collision anomalies are brief point events that contribute little to a window-level error signal; the dataset's original authors also found that the autoencoder-based detectors performed poorly on some of its subsets (Mascolini et al., 2023), suggesting reconstruction-based scoring has real limitations on this dataset. Future work should go towards averaging results across multiple random seeds in order to improve expected performance, and compare this approach directly against other unsupervised methods on this dataset.

References

- Alexandrov, A. (2024). LSTM-RNN method for anomaly-based intrusion detection systems. BISEC'2024: 15th International Conference on Business Information Security. CEUR Workshop Proceedings. <https://ceur-ws.org/Vol-3971/paper02.pdf>
- Alruwaili, A., Islam, S. M. N., & Gondal, I. (2025). Cybersecurity in robotic autonomous vehicles: Machine learning applications to detect cyber attacks. CRC Press. <https://www.routledge.com/Cybersecurity-in-Robotic-Autonomous-Vehicles-Machine-Learning-Applications-to-Detect-Cyber-Attacks/Alruwaili-Islam-Gondal/p/book/9781003610915>
- Canizo, M., Triguero, I., Conde, A., & Onieva, E. (2019). Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing*, 363, 246–260. <https://doi.org/10.1016/j.neucom.2019.07.034>
- Fosch-Villaronga, E., & Mahler, T. (2021). Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots. *Computer Law & Security Review*, 41, 105528. <https://doi.org/10.1016/j.clsr.2021.105528>
- IBM. (2025). Cost of a data breach report 2025. IBM Security. <https://www.ibm.com/reports/data-breach>
- IBM. (2026). X-Force threat intelligence index 2026. IBM. <https://www.ibm.com/reports/threat-intelligence>
- Kayan, H. (2025). Anomaly detection in industrial robotic arms using edge-based IoT systems [Doctoral dissertation, Cardiff University]. <https://orca.cardiff.ac.uk/id/eprint/179739/>
- Lu, H., Du, M., Qian, K., He, X., & Wang, K. (2022). GAN-based data augmentation strategy for sensor anomaly detection in industrial robots. *IEEE Sensors Journal*, 22(18), 17464–17474. <https://doi.org/10.1109/JSEN.2021.3069452>
- Mascolini, A., Gaiardelli, S., Ponzio, F., Dall'Ora, N., Macii, E., Vinco, S., Di Cataldo, S., & Fummi, F. (2023). Robotic Arm Dataset (RoAD): A dataset to support the design and test of machine learning-driven anomaly detection in a production line. *IECON 2023 – 49th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. <https://doi.org/10.1109/IECON51785.2023.10311726>

Park, Y., & Yun, I. D. (2018). Fast adaptive RNN encoder–decoder for anomaly detection in SMD assembly machine. *Sensors*, 18(10), 3573. <https://doi.org/10.3390/s18103573>

Zhou, K., Cao, Y., Kim, T., Zhao, H., Dong, H., Ting, K. M., & Zhu, Y. (2024). RAD: A dataset and benchmark for real-life anomaly detection with robotic observations. *arXiv*.
<https://arxiv.org/abs/2410.00713>

Zhou, X., Hu, Y., Liang, W., Ma, J., & Jin, Q. (2021). Variational LSTM-enhanced anomaly detection for industrial big data. *IEEE Transactions on Industrial Informatics*, 17(5), 3469–3477.
<https://doi.org/10.1109/TII.2020.3022432>