Michael Fouts
CPE 520
Final Report
December 11, 2023

An Application of Neural Networks to Predictive Maintenance

Introduction and Problem Statement

In an era marked by rapidly evolving industrial infrastructure, the demand for high reliability and minimal downtime is becoming increasingly more important. Companies face the challenge of maintaining complex systems, where unforeseen failures can lead to significant financial and operational setbacks. This paper delves into the potential of machine learning predictive models using neural networks as a proactive solution to anticipate and mitigate equipment failures.

When considering maintenance plans, there are three primary methodologies: Reactive, Preventative, and Predictive. Reactive maintenance, also known as "run-to-failure," is typically cost-effective in terms of capital and straightforward to implement but can lead to unpredictable downtimes and potentially higher long-term costs due to unplanned repairs. Preventative maintenance, typically thought of as scheduled-based maintenance, can be inefficient as it may lead to unnecessary maintenance activities. Predictive maintenance, utilizing real-time data and predictive analytics, is the most complex but offers significant long-term benefits by reducing downtime and extending equipment life. It allows maintenance to be performed just in time, enhancing efficiency and reducing costs associated with both reactive and schedule-based approaches. Visualizations of each can be seen in Figure 1: Example Maintenance Plans.

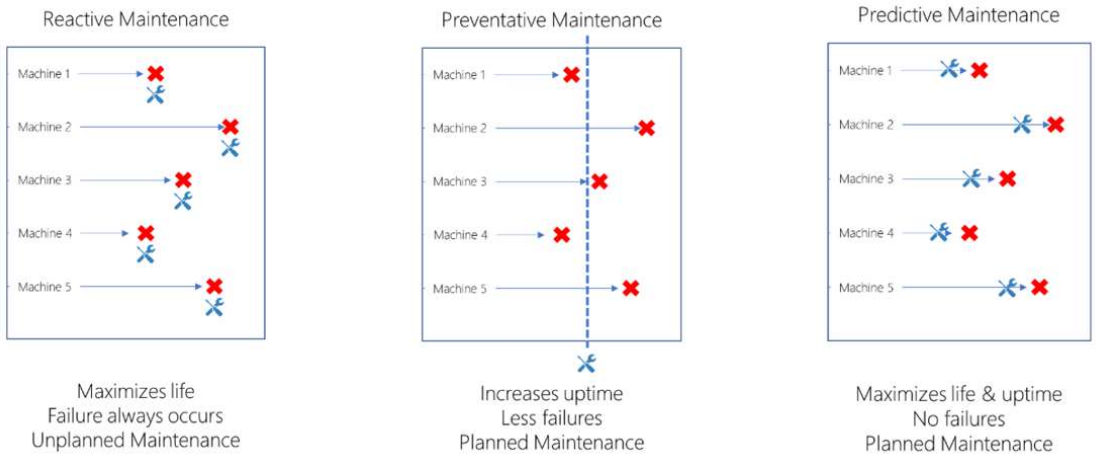

Figure 1: Example Maintenance Plans [1]

This paper explores the application of artificial neural network based predictive modeling techniques from Bangalore and Tjernberg (2015) [2] to forecast failures in machines. Bangalore and Tjernberg demonstrated successful early detection of wind turbine gearbox bearing faults using neural network modeling of SCADA temperature sensor data. Here, their nonlinear autoregressive neural network with exogenous inputs (NARX) method is adapted to model relationships between machine telemetry signals (provided by Microsoft Azure as a generalized, synthetic dataset[3]) and impending failures. To enhance this model, Long Short-Term Memory (LSTM) networks were explored for better prediction of telemetry signals and an economic analysis framework was integrated as the primary comparison of various scenarios. This multifaceted approach aims to focus on the primary factor in predictive maintenance, cost, to provide a more comprehensive economic understanding of potential predictive maintenance systems.

Background on Recurrent Models for Time Series Data

Recursive models are highly effective for time series predictions due to their ability to integrate past information into future forecasts. This is crucial in time series data, where historical patterns and trends often inform future behavior. By continuously feeding back their own outputs as inputs, recursive models can adapt and respond to the evolving nature of the data, capturing temporal dependencies and changes over time. This makes them particularly suited for forecasting in dynamic environments, where understanding and predicting the time-based sequence of events is key.

The NARX (Nonlinear AutoRegressive with eXogenous inputs) model is a type of recurrent dynamic network particularly suited for time-series forecasting. The model is made recursive by feeding its own predictions back as inputs to make further predictions. This architecture is designed to capture the nonlinear relationships between past values of a time series and other external, or exogenous, inputs to predict future values. By continuously updating model states, they can adapt to changing data patterns over time. This makes NARX ideally suited to applications like predictive maintenance, where the evolution of sensor measurements contains signals predictive of impending failures. The efficacy of NARX models in capturing complex temporal dependencies in data is well-documented in academic literature, making them a popular choice for time-series forecasting tasks.[4]

A key aspect requiring optimization is selecting appropriate input and output lags that capture relevant temporal dependencies without overfitting to noise. The input lag dictates how many past time steps are used as predictors, while the output lag determines the target prediction horizon. Choosing lags that align with inherent cycles and delays in the system dynamics can greatly improve forecast accuracy. Beyond neural networks, this general recursive approach of feeding predictions back as inputs can be integrated into other modeling techniques like regression or even statistical methods. An example of the NARX architecture applied with a neural network can be seen in Figure 2: NARX Neural Network Example [5].

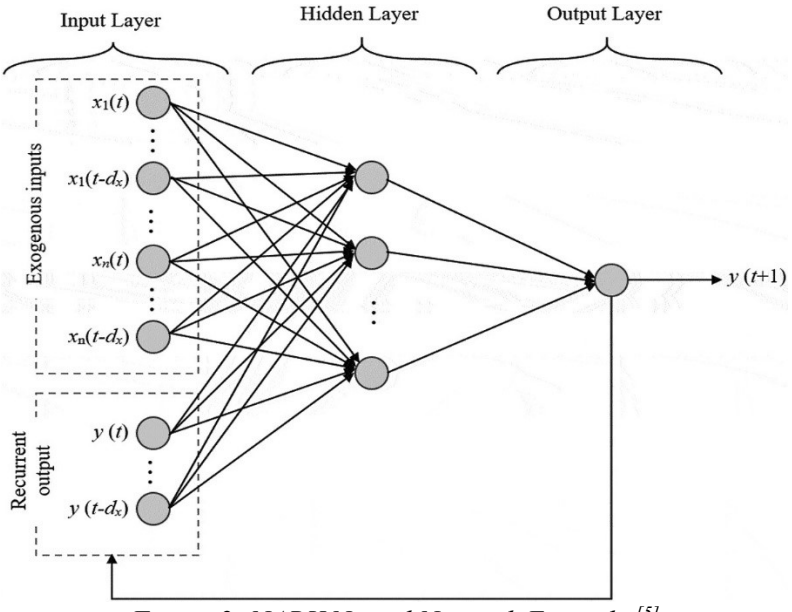

Figure 2: NARX Neural Network Example [5]

Long Short-Term Memory (LSTM) networks provide an alternative recurrent architecture that can effectively model longer-range temporal dependencies in time series data. Unlike basic recurrent neural networks, LSTM cells contain specialized logic gates including an input gate, output gate, and forget gate. These gates can select which information to remember, forget, or pass along to the next time step. This

gating mechanism gives LSTMs the ability to retain memories and learn patterns across longer sequences, mitigating issues with vanishing gradients.

Whereas NARX models take past output values as direct inputs, LSTMs maintain a cell state vector that encapsulates learned representations of historical context. The output gate then controls what is emitted based on this state. Both model types recursively pass information forward in time, but LSTMs can theoretically capture much longer range dependencies thanks to the self-looping cell state flow. This selectivity combined with a durable long-term memory enables LSTMs to hone in on pertinent patterns within noisy, high-dimensional time series data, making them an excellent candidate to explore as an improvement to this architecture.

Base Case Methodology

The original study's methodology involved three phases: data preprocessing, modeling, and statistical analysis for anomaly detection. To accommodate differences between the new dataset and that used by Bangalore and Tjernberg (2015) [2], adjustments were made in the current work. Specifically, the data was filtered to isolate a single machine, and successful failure prediction was redefined as detecting any warning sign within 48 hours prior to an actual error or failure event. This is because there are no traditional SCADA alarms or warnings to compare to like in Bangalore and Tjernberg (2015) [2]. By focusing the analysis on one machine and aligning the prediction target with available labels in the new dataset, the methodology could be adapted to evaluate the reproducibility of the previous results. One difference that still exists at this point though is that there are two different anomaly categories (errors and failures) in comparison to Bangalore and Tjernberg's one. This is addressed in the economic evaluation mentioned later on.

The time series nature of the data necessitated preprocessing it into continuous, equal-length timesteps before analysis. To accomplish this, the full dataset was first subset into sections based on the timestamps of equipment failures. The longest span of uninterrupted normal operation was identified, providing a robust training set of normal data. The 48 hours directly preceding each failure event were excluded to avoid training on potentially anomalous signals. With reliable normal data extracted, the dataset was then normalized on a 0 to 1 scale for each sensor variable to standardize the inputs. Finally, the normalized data was partitioned into a 90/10 ratio between training and validation sets, enabling optimization of the model's learning and predictive accuracy on previously unseen data.

For data modeling, an NARX architecture with a neural network was used, with pressure, rotation speed, and vibration measurements used as inputs to predict voltage within the machine. For the NARX model, the hyperparameter values were either taken from Bangalore and Tjernberg (2015) [2] or reasonable estimates were chosen. These parameters can be seen in Table 1: Base Case NARX Parameters.

Table 1: Base Case NARX Parameters

Parameter	**Value**
Input Timestep Lag	3, consecutive
Output Timestep Lag	3
Activation Function	Tanh
Number of Layers	3
Neurons per Layer	64
Error Function	MSE

The Mahalanobis distance was used as the anomaly detection metric for the NARX model's predictions. This metric accounts for correlations between variables by normalizing by the covariance matrix, making it well-suited for detecting outliers in multivariate data. Specifically, the two variables considered were the actual voltage and the error between the predicted and actual voltage.

The distributions of the Mahalanobis distances were then fitted to a two-parameter Weibull distribution to establish a threshold for anomalous results. The 99th percentile of the cumulative distribution function was set as the cutoff, flagging Mahalanobis distances exceeding this value as anomalies. This threshold was applied across the entire dataset for the machine to identify potential errors and failures. To best reflect real world behaviors, any anomaly flag occurring within 48 hours prior to an actual event was considered a correct prediction, regardless of frequency as this is expected to result in one maintenance activity.

Since the model does not differentiate errors from failures, a modified confusion matrix was constructed with both categories combined into a singular anomaly class. Owing to the imbalance between normal and anomaly classes, normalization was excluded from the confusion matrix to better gauge the magnitude of each category. Further processing was also required to account for multiple alerts preceding events based on the 48 hour prediction window defined for this analysis. The final results of the confusion matrix are shown in Table 2: Base Case NARX Confusion Matrix. These results are the best a confusion matrix can provide, but are still hard to interpret, leading to the economic evaluation.

Table 2: Base Case NARX Confusion Matrix

	Anomaly (Prediction)	Normal Operations (Prediction)
Anomaly (True)	$\frac{17}{40}$	$\frac{23}{40}$
Normal Operations (True)	$\frac{96}{7206}$	$\frac{7110}{7206}$

Economic Evaluation

The confusion matrix results proved challenging to interpret and compare across models, as some categories might improve while others worsen. To enable clear evaluation, an economic model was developed to assign a single monetary value representing overall performance. This required denominating each possible prediction outcome. Table 3: Maintenance Economics Table was used for the economic assessment, with values based on estimates for maintenance costs of datacenter servers, since the dataset was originally published by Microsoft Azure and is generalized with no specific use case.

Table 3: Maintenance Economics Table

Result	**Cost**
Correctly predicted Failure	$50
Incorrectly predicted Failure	$500
Correctly predicted Error	$25
Incorrectly predicted Error	$50
False Positive (Unnecessary Maintenance)	$10
Correctly Predicted Normal Operations	$0

With the values given by the table, the overall base case cost for a year of operation was $3,560.

Optimized NARX Model

Having established the economic model for comparative evaluation, the next phase focused on hyperparameter optimization to minimize maintenance costs for the machine. While multiple optimization areas were targeted, three had disproportionate significance and will be focused on for this report: the NARX model's input/output lag, the Mahalanobis distance anomaly threshold, and the neural network architecture itself.

Selecting appropriate input/output lags is critical in time series forecasting to capture underlying data patterns. Too short and important dynamics may be overlooked; too long and excessive noise is introduced. Both degrade prediction accuracy. Optimization resulted in an output lag of 5 steps, with input lags of 1, 2, 3, 12, 24, and 36 steps. The non-consecutive inputs aimed to incorporate daily cyclical patterns while avoiding noise from intermediate readings.

Though not originally tuned in the baseline method, the Mahalanobis threshold proved impactful in reducing false positives, and was thus included in optimization. To avoid bias, a secondary training set supplemented the NARX data. Optimization yielded the 99.5th percentile as the optimal CDF cutoff, significantly curtailing non-actionable alerts. An important note to mention is that for this optimization in particular, a different training dataset is needed, particularly one with anomalies in it. The entire dataset was split 50/50 for training and validation due to the sparse nature of anomalies.

Lastly, the neural network architecture was optimized. While this contains several hyperparameters, dominating considerations were the loss function and the number of neurons in each layer. Based on similar networks from literature, a maximum of three layers was chosen due to the limited complexity of the input space and minimum improvement in prediction capability.

A complete table of hyperparameters can be found in Table 4: Optimized Case NARX Parameters.

Table 4: Optimized Case NARX Parameters

Parameter	**Value**
Input Timestep Lag	Lags of 1,2,3,12,24, and 36
Output Timestep Lag	5
Activation Function	Leaky ReLU
Number of Hidden Layers	3
Neurons per Layer	128
Error Function	MSE
Mahalanobis CDF Threshold	99.5%

The overall cost of this model was $2,720, a 24% reduction from the base case largely due to a much higher accuracy in anomaly detection with only minor additions of false positives.

LSTM Predictions

LSTM networks were investigated for predicting voltage (referred to as the Single Output Case), given their prowess in processing temporal sequences. However, the LSTM failed to capture the dynamics as accurately as the optimized NARX model, with a higher MSE of 0.02 vs 0.016. This implies the NARX architecture provided a better fit to the intricacies of this dataset. Still, the LSTM model improved economics by $425 compared to baseline.

A key advantage of LSTM is facilitating multivariate forecasting. This allowed for a different methodology to be assessed where all four sensor metrics were predicted simultaneously, with the

Mahalanobis distance recalculated using errors across all outputs (referred to as the Multiple Output Case). However, this approach performed similarly to just modeling voltage alone. So while LSTM warrants future exploration, the complex univariate NARX approach was better suited for this initial reproducibility analysis. All parameters for the LSTM models can be found in the Appendix of this report.

Other Optimization Attempts

Additional techniques were explored to further optimize performance, though ultimately did not improve on the results discussed previously. Given the limited input space complexity, input transformations were applied to the NARX model to capture potential non-linear relationships. Both 2nd degree polynomial and Fourier kernels were implemented. However, these added excessive noise that degraded predictions, implying the raw data itself provided optimal model fitting.

Separately, data from other machines with longer stretches of normal operating data was blended to expand the training set size. The concept was that learnings from similar units could transfer. However, models trained on aggregate data performed worse when tested on the target machine than using its data alone. This suggests machine-specific models may be necessitated for production-level deployment, rather than a one-size-fits-all approach.

Results

When comparing the methodologies against one and other, it is found that the predictive maintenance approach would be more cost effective than other existing maintenance methodologies, such as reactive maintenance, and that the optimized NARX architecture is the most cost effective. Results can be seen in Figure 3: Architecture Cost Comparison.

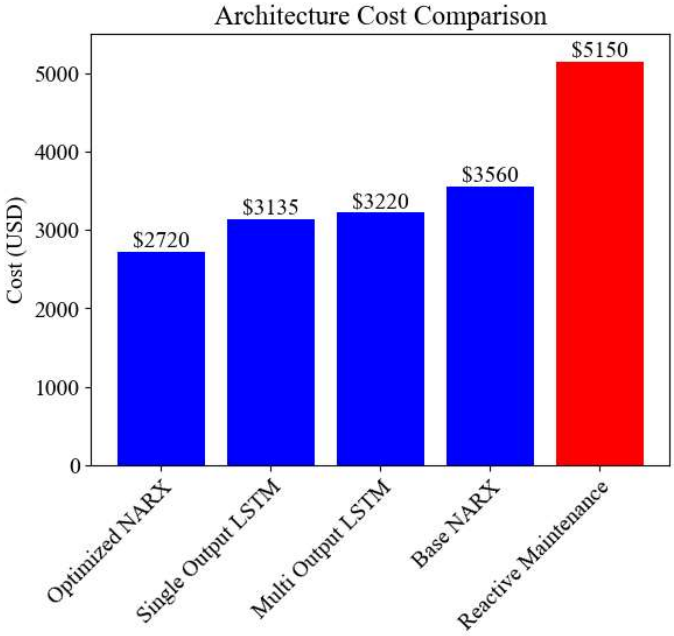

Figure 3: Architecture Cost Comparison

Conclusion

In conclusion, this research has demonstrated the effectiveness of predictive maintenance methodologies, particularly the optimized NARX model, in reducing operational costs. The comparative analysis with

LSTM and other models highlights the capability of NARX in predicting system dynamics, achieving lower error rates and economic benefits. These findings underscore the potential of machine learning in predictive maintenance, offering a more cost-effective approach than traditional methods. The study not only contributes to the field of predictive maintenance but also opens avenues for further research in optimizing machine learning models for industrial applications.

All code for this project can be viewed at **github.com/michaelwfouts/predictive-maintenance-project**.

References

[1] "Predictive Maintenance Overview," Microsoft Azure Architecture Center. [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/industries/manufacturing/predictive-maintenance-overview. [Accessed: 9-Dec-2023].

[2] P. Bangalore and L. B. Tjernberg, "An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings," in *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 980-987, March 2015, doi: 10.1109/TSG.2014.2386305.

[3] A. Biswas, "Microsoft Azure Predictive Maintenance," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance. [Accessed: 9-Dec-2023].

[4] Diaconescu, Eugen. "The use of NARX neural networks to predict chaotic time series." Wseas Transactions on computer research 3.3 (2008): 182-191.

[5] W. H. AlAlaween et al., "A dynamic nonlinear autoregressive exogenous model for the prediction of COVID-19 cases in Jordan," Cogent Engineering, vol. 9, no. 1, 2022, doi: 10.1080/23311916.2022.2047317.

Appendix

*Table 5: LSTM Parameters, Single Input**

Parameter	**Value**
Input Timestep Lag	9, consecutive
Activation Function	ReLU
Number of LSTM Units in First Layer	64
Number of Fully Connected Hidden Layers	3
Neurons per Fully Connected Layer	64
Error Function	MSE
Mahalanobis CDF Threshold	99.5%

*Table 6: LSTM Parameters, Multiple Outputs**

Parameter	**Value**
Input Timestep Lag	9, consecutive
Activation Function	ReLU
Number of LSTM Units in First Layer	64
Number of Fully Connected Hidden Layers	3
Neurons per Fully Connected Layer	64
Error Function	MSE
Mahalanobis CDF Threshold	99.5%

*Note: Both LSTM models were trained separately, but were found to have the same parameters