Anomaly Detection Using Deep-Learning Models on Sensor Data

Abstract—**The reliability and availability of critical assets such as machinery, motors, and pumps are essential for smooth operations in heavy industries. Failure of these assets can lead to significant financial losses. Asset management programs are put in place to ensure the integrity and reliability of these assets. Anomaly detection plays a critical role in the Asset Management program, as it allows for the early detection of anomalies in equipment behavior, mitigating risks and preventing unplanned downtime. In this paper, we propose using anomaly detection techniques to detect anomalies in the sensor readings of a pump and predict potential failures. By analyzing the sensor readings from the pump, we can identify patterns and anomalies that could help predict potential failures. We can use machine learning algorithms to develop an anomaly detection model that can detect anomalies in real time and provide timely alerts to maintenance personnel. In this project, we explore two different aspects of the problem: anomaly detection and anomaly prediction (classification). For the first task, we address it with an Autoencoder architecture implemented with Pytorch. The second problem we approach it with a Deep Neural Network.**

Index Terms—**Outlier detection, Autoencoders, Multi-Layer Perceptron, Sensor Data**

I. INTRODUCTION

In heavy industries such as manufacturing, machinery, motors, and pumps play a critical role in ensuring smooth operations. Any failure of these assets can result in production loss, leading to significant financial losses. Asset management programs are put in place to ensure the integrity and reliability of these assets, with highly skilled reliability engineers managing the process. Anomaly detection, a subfield of machine learning, plays a crucial role in the Asset Management program, as it allows for the early detection of anomalies in equipment behavior, mitigating risks, and preventing unplanned downtime.

One of the significant challenges in the industry is the early detection of anomalies in equipment behavior. Anomaly detection techniques are essential to identify patterns and anomalies that could help predict potential failures. The problem of detecting anomalies is particularly challenging in the context of sensor data, where the sheer volume of data generated makes it challenging to analyze manually.

Deep neural networks (DNNs) have been widely used for anomaly detection tasks in recent years. They are a popular choice due to their ability to learn complex representations of data and their flexibility in handling large datasets. In anomaly detection, DNNs can be used as a classification problem, where the network is trained to classify data as either normal or anomalous. The network is first trained on a dataset of normal data, and then during testing, it is evaluated on new data to determine if it is normal or anomalous.

One of the challenges in using DNNs for anomaly detection is the lack of labeled anomalous data, which is often rare or difficult to obtain. One approach is to use autoencoders, a type of DNN that learns to reconstruct input data from a compressed representation, as an unsupervised anomaly detection method. An autoencoder is trained to reconstruct normal data, and during testing, if the reconstruction error is higher than a certain threshold, the data is classified as anomalous. This approach has shown promising results in various anomaly detection tasks, including intrusion detection, fraud detection, and equipment failure prediction.

Autoencoders, a type of neural network, are increasingly being used for anomaly detection in sensor data. They can learn to encode the patterns present in the sensor data and can detect deviations from these patterns, which may signify anomalous behavior. The use of autoencoders for anomaly detection has shown promising results in various industrial applications. In this paper, we explore the use of autoencoders for anomaly detection in pump sensor data and evaluate the effectiveness of the proposed approach in detecting anomalies and predicting potential failures.

II. RELATED WORK

Anomaly detection is a critical problem in machine learning, and there has been a significant amount of research done in this area. Many different techniques have been proposed, including statistical methods, machine learning algorithms, and deep learning models. In recent years, autoencoders have gained popularity as a powerful tool for anomaly detection, particularly in high-dimensional data.

One of the early papers on anomaly detection using autoencoders was proposed by [1]. They introduced a deep autoencoder model that could learn a low-dimensional representation of high-dimensional data, and the model could detect anomalies by measuring the reconstruction error between the input and output data. They demonstrated that their method outperformed traditional statistical methods on several benchmark datasets.

Another significant contribution to the field of anomaly detection using deep learning is the work done by [2]. They proposed a novel deep autoencoder model that used convolutional layers to learn local features in medical images. They demonstrated that their method outperformed traditional methods for detecting anomalies in medical images.

Another recent paper by [**?**] proposed a deep autoencoder model with a novel loss function for detecting anomalies in time-series data. They demonstrated that their method could outperform traditional methods on several benchmark datasets, including the well-known KDDCup99 dataset.

In addition to autoencoders, deep learning models have also been used for anomaly/outlier detection in other ways. [3] proposed a deep learning model that could detect anomalies in graphs using a novel loss function based on graph embeddings. They demonstrated that their method could outperform traditional methods on several benchmark datasets.

Another recent paper by [4] proposed a deep learning model for anomaly detection in images that used a contrastive loss function. They demonstrated that their method outperformed traditional methods on several benchmark datasets, including the MNIST dataset.

In recent years, deep learning models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have shown promising results for anomaly detection in time series data. [9] proposed a deep autoencoder model for detecting anomalies in multivariate time series data. The model learned a low-dimensional representation of the data and used reconstruction error to identify anomalies. They demonstrated the effectiveness of their model on several benchmark datasets, including the Yahoo S5 dataset and the NASA turbofan engine dataset.

Another recent work by Akhtar et al. (2018) [10] proposed a hybrid approach for anomaly detection in time series data, which combined the strengths of traditional statistical methods and deep learning models. They used a hybrid LSTM-ARIMA model for time series forecasting and detected anomalies by comparing the forecasted values with the actual values. They demonstrated the effectiveness of their model on the Numenta Anomaly Benchmark (NAB) dataset.

In conclusion, autoencoders and deep learning models have shown great promise in the field of anomaly detection, particularly in high-dimensional data. The above mentioned papers demonstrate the effectiveness of deep learning models in detecting anomalies in various types of data, including images, time-series data, and graphs.

III. Methodology

The dataset used in this study is the Pump Sensor Data, which is a time-series dataset collected from a sensor attached to a pump in an industrial setting. It consists of 52,170 rows and 52 columns, with each row representing a time-step and each column representing a sensor reading. The dataset was obtained from Kaggle and can be found at [5].

The labels for this dataset consist of Normal, Recovering, and Broken. The class distribution can be seen in the figure below. As we can see from the plot, there are far fewer negative class data points than the normal data points. This will pose a problem when training a classifier.

Before the dataset could be used for training the models, preprocessing was required to ensure the quality of the data. Firstly, any rows or columns with missing values (NaN) were

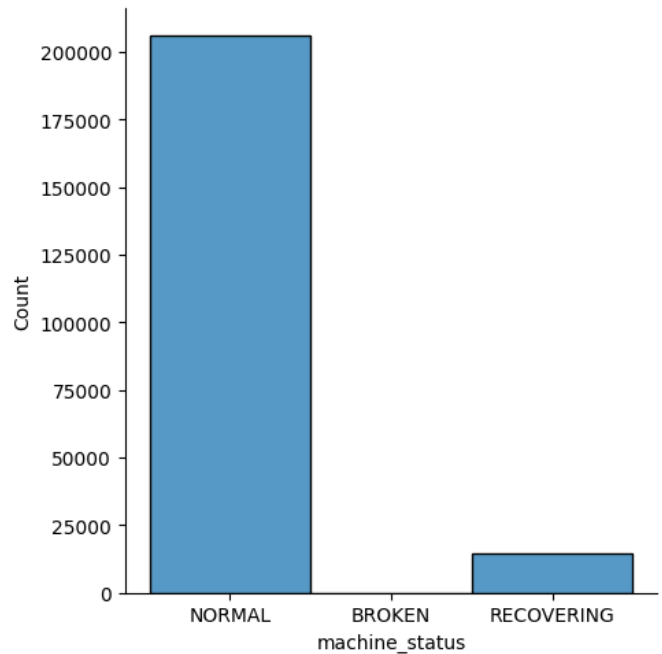

Fig. 1. Class Distribution.

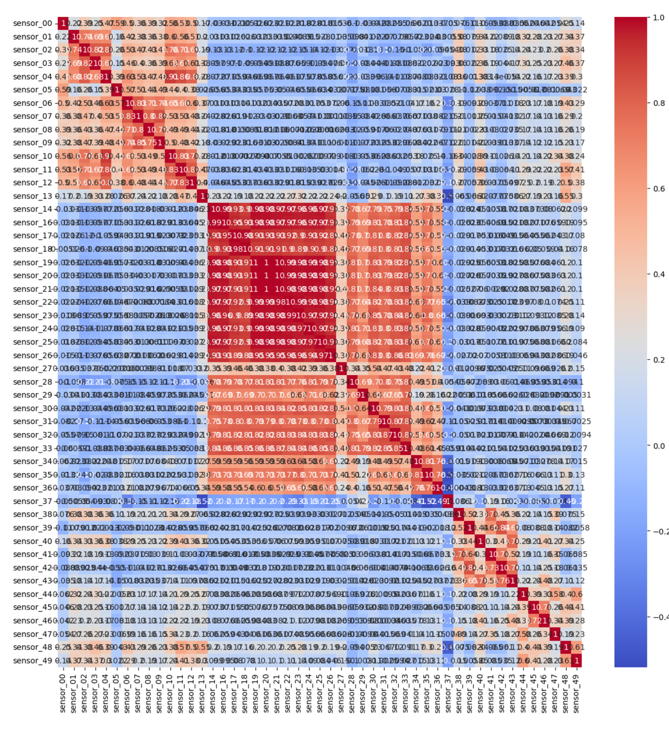

Fig. 2. Correlation between features.

removed from the dataset with this process we elimanted a few columns. Then, interpolation was used to fill in any remaining missing values. Finally, the dataset was scaled using min-max scaling to ensure that each feature has the same range and to prevent any one feature from dominating the model.

The data for both of the models were prepared by a sliding window with a sequence length of 60. This ensured that the model would capture the complete aspect of the data and would give us more training examples.

Two models were used in this study for anomaly detection: autoencoders and deep neural networks. Autoencoders were implemented using Pytorch Lightning [7], a high-level interface for Pytorch that simplifies the training and deployment of deep learning models. A custom Dataset class was created to load the data, and the model was trained on a single GPU. The autoencoder was trained only on positive examples to learn the data. After the training, an average reconstruction error was calculated on the training data. This value was then used as a threshold for anomaly detection, where any instance with a reconstruction loss higher than the threshold was classified as an anomaly. For monitoring the training process, we used Weights and Biases, a tool that helps you explore your CPU/GPU usage, train/test/validation loss and many more functionalities to help you track your model training [6].

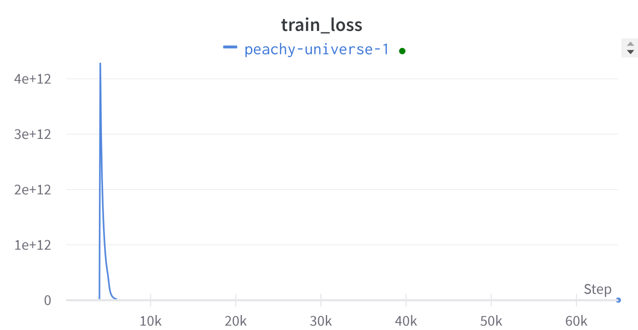

Fig. 3. Train loss on Autoecnoder.

The architecture of the autoencoder consisted of an encoder layer (60 * 49 > 25 > Relu > 15) and a decoder layer (15 > Relu > 25 > 60 * 49). We trained the model for 10 epochs, using Mean Squared Error loss and Adam Optimizer with a learning rate of 0.001. The model had 13.3 K trainable parameters.

Deep neural networks were also used in this study to predict if there would be an anomaly in the next 20 minutes based on a sequence length of 60 seconds (60 data points). Initially, data imbalance was a problem, but this was resolved by using a balanced dataset where the created dataset consisted of positive examples if the next 20 minutes there was a recovering or a broken machine. The neural network architecture consisted of two hidden layers with 200 and 40 neurons, respectively, and a single output neuron with a relu activation function. For this task, the Machine Learning Library TensorFlow [8] was used.

```
  | Name    | Type       | Params
---------------------------------------
0 | encoder | Sequential | 6.5 K
1 | decoder | Sequential | 6.8 K
---------------------------------------
13.3 K    Trainable params
0         Non-trainable params
13.3 K    Total params
```

Fig. 4. Autoencoder architecture.

We trained the model for 150 epochs on 800 positive and 800 negative samples. We used Stochastic Gradient Descent as our optimizer and Mean Square Error as our loss function. The model had 600 K parameters.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 200)               588200

 dense_1 (Dense)             (None, 40)                8040

 dense_2 (Dense)             (None, 1)                 41

=================================================================
Total params: 596,281
Trainable params: 596,281
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5. Deep Neural Network - Classification model.

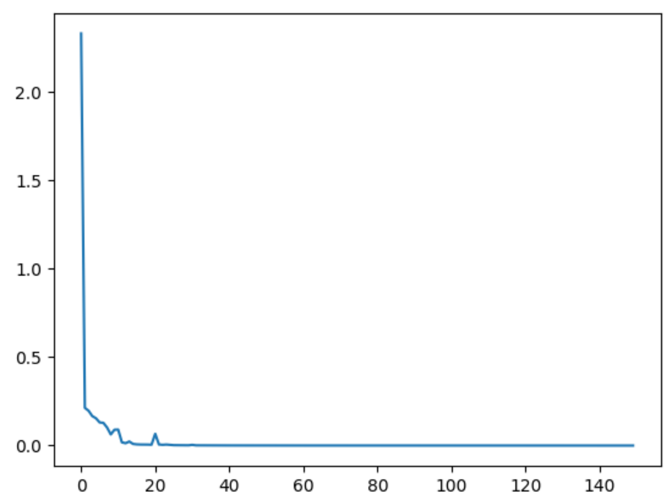

Fig. 6. Deep Neural Network - Classification model.

In summary, the dataset was preprocessed to ensure data quality, and two models were used for anomaly detection: autoencoders and deep neural networks. Autoencoders were trained only on positive examples and used the average reconstruction loss as a threshold for anomaly detection. Deep neural networks were used to predict if there would be an anomaly in the next 20 minutes based on a sequence length of 60 seconds.

IV. Evaluation and Results

The Autoencoder model did not perform well in detecting anomalies in the sensor data. Although the model showed promising results during training, it did not generalize well on the test data. One possible reason for this is the lack of negative samples used in training the model. Since the autoencoder was only trained on positive data, the reconstruction error for negative examples was not significantly different from normal data, making it difficult for the model to detect anomalies accurately. While this issue can be addressed by lowering the threshold for anomaly detection, it may increase the risk of false positives, which can be costly. Thus, further investigation is required to determine the optimal threshold for detecting anomalies using the Autoencoder model.

In contrast, the deep learning model yielded highly promising results in detecting anomalies in the sensor data. By predicting whether a machine is going to break down in the next 20 minutes, the model achieved zero false negatives, indicating that it can accurately predict potential failures. The confusion matrix obtained from the model also supports its efficacy in detecting anomalies with high precision and recall values. These results suggest that the deep learning model is a promising solution for detecting anomalies in industrial machinery, which can ultimately reduce the risk of unplanned downtime and mitigate financial losses.

TABLE I
Confusion matrix

	Actual Positive	**Actual Negative**
Predicted Positive	201	155
Predicted Negative	0	546

V. Conclusion And Future Work

In conclusion, we have explored and compared two approaches to anomaly detection in sensor data from pumps. Our deep learning model was successful in detecting all anomalies, achieving 0 False Negatives, and provided a reliable classification of potential equipment failure in the next 20 minutes. On the other hand, our autoencoder model did not meet our expectations, mainly due to the lack of negative samples during training. However, we believe that this approach still holds promise and could be further improved with a larger dataset that includes negative samples.

For future work, we suggest exploring different types of deep learning models, such as recurrent neural networks or convolutional neural networks, to improve the performance of the classification model. Additionally, we could investigate methods to address the imbalance in our dataset, which could further improve the performance of our models. We also recommend exploring the use of unsupervised anomaly detection techniques, which could be more appropriate in cases where labeled data is scarce. Lastly, it would be valuable to apply our approach to real-time monitoring of pump sensor data in an industrial setting to evaluate its effectiveness in detecting equipment failure and preventing unplanned downtime.

References

[1] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[2] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., & Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In International conference on information processing in medical imaging (pp. 146-157). Springer, Cham.

[3] Golan, I., & El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In International conference on machine learning (pp. 1677-1686). PMLR.

[4] Bergman, L., & Hoshen, Y. (2020). Classification with Contrastive Loss and Regularization in the Absence of Labels. arXiv preprint arXiv:2005.11867.

[5] https://www.kaggle.com/datasets/nphantawee/pump-sensor-data.

[6] https://wandb.ai/site

[7] https://www.pytorchlightning.ai/index.html

[8] https://www.tensorflow.org/

[9] Malhotra, P., Vig, L., Shroff, G., Agarwal, P. (2016). LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148.

[10] Akhtar, M. Z., Gilani, S. Z., Song, H. (2018). Hybrid lstm-arima model for anomaly