HALMSTAD UNIVERSITY

Masters in Embedded and Intelligent Systems

MASTER THESIS

Smart Alarm -Alarm Prediction using sensor data

Ahamed Buhari, Chandraprakash Sahu

Information Technology

Halmstad University, August 24, 2021–version 1.4

Ahamed Buhari,Chandraprakash Sahu: *Smart Alarm-Alarm Prediction using sensor data,* , © June 2020

SUPERVISORS:
Hadi Fanaee , Mahmoud Rahat

LOCATION:
Halmstad, Sweden

CONTENTS

ACRONYMS

PM Predictive Maintenance

CNN Convolutional Neural Networks

LSTM Long-Short Term Memory Neural Network

MAE Mean Absolute Error

MSE Mean Square Error

1 INTRODUCTION

1.1 BACKGROUND

The Introduction of information-related data advancements in manufacturing enables the collection of massive data from the sensors observing the creation measures and various alerts. Early detection of these alarms can provide several economic benefits to manufacturing companies, including predictive maintenance of equipment and production optimization.

Time series forecasting is an important area of machine learning that is often neglected. It is significant because numerous expectation issues include a period segment. These issues are dismissed because this time part makes time-series problems harder to deal with. The Analyzers can distinguish cautions that can be demonstrated with basic guidelines dependent on the initiation condition and more mind-boggling cautions. It is obscure when the enactment condition will be satisfied. The company provides the data used in this project. We will work with a sub-sample of data retrieved from Alfa Laval's IoT portal, corresponding to 27 water purifier machines installed on different ships. These machines purify the oil and water supplies onboard marine vessels. In absolute, we have 102 sensors, and 115 parameters set by the client could be altered under various circumstances. The information is gathered for a time of 3-8 months, contingent upon the machines. The company is keen on developing an intelligent alarm system that predicts alarms and warnings before their event, dependent on sensor readings. Thus, the principal objective of our venture is to anticipate cautions and warnings utilizing AI methods and track down the best prescient models.The objective is to exploit the robust model,to be implemented on the machine's controller to make real-time predictions. As of now, these cautions are working dependent on a predefined edge limit, and once it crosses the cutoff, the framework creates deficiency and cautioning messages. On the other hand, predictive models will improve the machines' helpful existence, evade vacation, upgrade the item quality and amounts, and forestalls damages or faults. Deep learning methods like Neural networks are the state-of-the-art algorithms used for working with time-series data. They can learn the temporal dependence from the data. However, the input of these algorithms has to be very large. When the data available is less, there is scope to investigate how we can improve the performance of machine learning models.

1.2 PROBLEM FORMULATION

The company is interested in developing a smart alarm system that predicts alarms and warnings before their occurrence based on sensor readings. So, the main objective of our project is to predict alarms and alerts using machine learning techniques and find the best predictive models. The goal is to exploit the robust model,to be implemented on the machine's controller to make real-time predictions. Presently, these alarms are functioning based on a predefined threshold limit, and once it crosses the limit, the system generates fault and warning messages. Predictive models will improve the machines' useful life, avoid downtime, enhance the product quality and quantities, and prevent damages or faults.

1.3 RESEARCH QUESTIONS

RQ1: What is the ideal window size for taking samples before the occurrence of alarms and warnings?

RQ2: What sampling frequency (e.g., 5 seconds, 1 minute, 5 minutes) in time series is ideal for making better predictions?

RQ3: Which sensors and parameters contribute more for making better predictions?

RQ4: Which machine learning method provides better predictions (e.g., classical or sequential)?

RQ5: Which set of configurations makes a better predictive model? We will investigate different machine learning techniques and find which gives the better model. In particular, we would like to compare regression algorithms with deep learning-based neural network models such as CNN and LSTM.

.

1.4 NOVELTY

This is an experimental research project with a high industrial impact. So far, the company has not used any predictive models on its data. So, making accurate predictions can provide added value to machines and provides profits for the company and the customers. Unfortunately, currently, there is no off-the-shelf machine software for direct machine learning on this type of data (multi-sensor unequal-length time series). We are experimenting with Tensor regression method with Hankelization and one more dimension" Lag" to check model performance and comparing with Vector and Matrix methods On linear and Neural Network models. Also, finding the ideal sampling frequency and other configurations such as window size, lag, which contribute to efficient model performance on twenty-seven machines.

Based on previous experiments done by other researchers related to similar type work, We have a hypothesis that out of these experimenting methods, Tensor regression method should outperform than matrix and vector regression methods.

1.5 MOTIVATION

Machines and product equipments maintenance are most crucial and challenging task for an industry to maintain consistent productivity, Product quality, delivery time and safe working environment. Effective maintenance brings lots of benefits for the industry majorly operational cost reduction, equipment life increases, production efficiency increases, machine fault reduction, delivery on time and many more.[15] Based on previous studies, a company estimate 80 percentage of their technician work on reactive maintenance issues than performing preventive maintenance procedures and the company can save up-to 18 percentage on average in cost by performing preventive practice.[8]

.

The industries are going through The Fourth Industrial Revolution and transforming and acquiring best for their Industries. Most of the industries including Alfa Laval company are working on conditional based maintenance to avoid from the reactive maintenance problems. The conditional based approach performed at the moment when the measure parameter of a machine reach to threshold level obtained through measurement systems. However, there are several disadvantages on this existing based approach and most vital is, it does not give enough time to plan and to line up right skill, parts and tools because the method works on real time. To overwhelm this problem Predictive maintenance using machine learning approach is most suitable

.

Predictive maintenance use monitored data to predict machine condition in future and give enough time to make decision based on prediction. Sometimes new working conditions such as new manufactured component, new material and etc are not suitable to set the equipment values permanently which inability to provide the correct machine status, another problem is one feature may influence the other features which can not be controlled perfectly using the existing condition based approach. However, Machine learning predictive maintenance based approach will minimise all these problems and provide the best accurate predicted values to help company to reach to the goal of Near-Zero breakdown/failure. Mostly, all the industries have goal to near zero breakdown, which can be possible with this approach.

In this thesis,we want to make predictions in alarms not only based on corresponding explicit signal but also from other relevant signal before the occurrence. For this reason, we are examining how we can model the auto correlation in multivariate time series in different way with tensor representation. We are inspecting whether tensor representation with multivariate time series provide better prediction performance. Moreover, we are also comparing this method with the traditional multivariate time series method and then comparing with modern techniques like CNN, LSTM to see how the tensor representation affect the prediction performance.

2 LITERATURE REVIEW

From the application perspective, the thesis is related to the area of research called predictive maintenance. In this paper,Zhang et.el. [17] center around data-driven strategies for Predictive maintenance, present an exhaustive study on its applications. According to methodological perspective, this postulation is about (multivariate) time series [11] and semi-supervised anomaly detection Cook et.el.[3]. In this way, frames the connected work in the extended term incorporates any results related to these areas.

Anomaly detection deals with the identification of events that gives rise to uncertainties due to a substantial deviation. The methods for anomaly detection can be classified into supervised and unsupervised approaches. In supervised methods, the model is fed with labeled data of both normal and abnormal instances, and then the model can be used for predicting peculiar instances.

On the contrary, unsupervised methods do not use labeled data, and they can identify anomalies by just comparing them with the significant part of the data, which is assumed to be expected. One of the most naive solutions in unsupervised anomaly detection for conditioning monitoring is to use statistical process control techniques on the sensor measurements along with a threshold limit. For instance,[7] the authors used this method for anomaly detection in the manufacturing process.

A. Horelu et al.[4] discussed the forecasting techniques for the time series from the sensor data. Uses Machine learning methods for forecasts on time series and pick the best models that fit our utilization case, and ready to anticipate future qualities dependent on recently accumulated information ends up being significant. By having the option to forecast the following values or values of a time series, we can estimate with the degree of certainty we have in the model that created the forecast how a specific environment variable may advance, and we can make the right moves. To predict the next value in a time series, we first train a model for that time series and then use that model to predict the next observed state.

WH Wan et al.[6]explain the importance of the window sliding technique, which compares the time delay within the dataset. The sliding window strategy was demonstrated capable of distinguishing the patterns from the temporal data. The datasets discussed here represent the different sliding sizes. Each sliding size means the time duration of the delays. The research idea emphasizes comparing window sizes and instances for different datasets and selecting the streamlined results. The neural network is taken as a modeling tool in this paper[6]. The model is compared with each datasets of different settings, and this procedure aims to get the combination that gives the best-optimized results.

Asante-Mensah et al.[2] stated the significance of hankelization in tensors. He further added that the fruition of data tensor with structured missing segments is quite a complex task. The main idea is to hankelize the deficient data tensor to acquire the high order tensors. This Hankelization is used to duplicate the patches of an image with prescribed window sizes. Even though it is feasible to utilize more refined Hankelization, relatively simple row hankelization procedures are performed due to their simplicity and efficiency. Each row of the frontal slices of the tensor data is hankelized using a given window size. Then, Hansel frameworks are built from these lattices block and having computed all the block Hankel matrix; the higher-order tensors are reshaped.

Newman, Elizabeth et al.[10] proposed a tensor neural network system that offers an energizing new worldview for designing neural organizations with multi-dimensional (tensor) data. The author here introduces the neural network armature, which replaces matrices with tensors. Tensor decompositions can remove more significant highlights when information usually is high-dimensional. Some new neural network models utilize tensor-train layers to separate multi- dimensional data. This system applies an arrangement or train of center frameworks to every component of an input tensor and returns one element of an output tensor.

Algorithmically, this tensor-train layer lessens the number of learnable boundaries in the network and can be executed effectively. Like other tensor approaches, the quantity of the learning parameters is reduced due to the high dimensional structure. Thus the authors conclude by defining that the t-product based tensor neural organization is a regular multi-dimensional augmentation of conventional neural networks. In this high-dimensional system, we can encode more data with fewer boundaries.

Taufik Nur et al[1] explained the automatic false alarm labeling for sensor data and proposed a strategy that emulates how the domain expert determines the bogus cautions. The authors explicated sensor data with few operations to determine whether the alarm is true or false. However, there are few cons in the process. To overcome this, the authors introduced a new method to determine false alarms. Ineffectual alert administration caused by false alarms can lead to grave misfortune. Based on the trial results[10], the authors inferred that the proposed strategy could effectively identify the bogus cautions and label sensor data. The labeling of sensor data is impacted by few factors, such as lack of consideration of the machine's state. By the interaction of sensor information logging, the outcome being that the sensor esteems are excessively scattered. The outcomes introduced here are restricted to offline sensor information marking, and the portrayal of the anomalous just dependent on one sensor information, not various sensor information. Nonetheless, this technique is helpful as a benchmark for further advancement of programmed false alarm detection to expand the security activity in the industry.

Hou et al.[5] discusses the various viewpoints concerning high-tensor order, including the fundamental ideas of tensor polynomial math and its deterioration. For the tensor rudiments, the authors present the essential documentation and definitions about tensor and their comparing tensorial tasks, concerning the tensor decomposition designs, the focus shifts around the most broadly utilized CP model and Tucker model. They described how to scale up CP and Tucker decompositions to deal with large-scale tensorial data and apply the tensor decomposition to big data applications; several approaches are designed to overcome scalability and efficiency. Compression, sparsity, tensor networks are the few strategies are discussed to scale up the tensor decomposition. Hence by this approach, the enormous scope of higher data information can be roughly addressed in exceptionally packed and dispersed configurations, bringing about both upgraded translation and computational benefits. The Figure[1]illustrates the representation of tensor.Tensor is a multi-dimensional array that represents a generalization of vectors and matrices. It is a multi-dimensional array with several dimensions and has a collection of integers arranged on a regular grid with a variable number of axes in the general case.[12].The dimension of the tensor is called a rank. The main difference is that tensor is the generalized form of scalars and vectors; it means scalars and vectors are the special cases of tensor quantities. Scalar is a tensor of rank 0, and vector is a tensor of rank 1. Low-rank tensor regression is a class of supervised learning models, and the main objective is to learn high-order correlations. Tensor regression has shown to be advantageous in learning tasks with multi-directional relatedness.

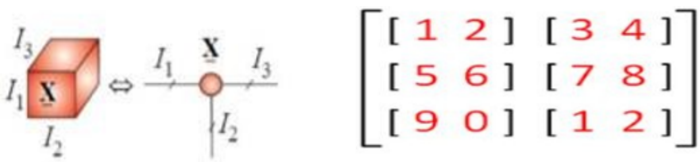

Figure 1: Tensor Representation

To accurately model temporal inputs in sequence labeling the authors, [12] stated there should be another construction to deal with the new dimension of time. The neural network model cannot recognize the time, making it imperfect for consecutive issues. In any case.Convolutional networks are discussed here to address this issue.

A variety of CNN could be applied to temporal data. The learned boundaries in CNNs are predefined windows that play out a convolution on cuts of information. As this window pushes through our info, the convolutions produce an intellectual component of the given window. Consequently, CNNs are utilized on consecutive information to extricate highlights from a window of tokens used by higher layers for expectation.

3 METHODOLOGY

In this chapter, we are presenting our proposed method. The chapter is divided into two main sections: Exploratory Data Analysis and Architecture.

3.1 DATASETS

The time-series datasets are referred to as the datasets that have the time aspect involved in them. These kinds of datasets are generated by collecting the data at different points in time. Data selection is one of the important and challenging part, and once the data is gathered, time series analysis is performed to understand past performance of data to train the model and identify the similar pattern for future prediction.

The original data was in time series of different sensors, the data is selected for one hour from 5 seconds before the point when the alarms or warning happens until one hour before of it. Then the combined samples of all sensors data for each machine, generates tensor data. Further, the data is normalised and use for training and modelling purpose. Moreover,we are also ignoring the alarm which does not have enough data samples in time window.

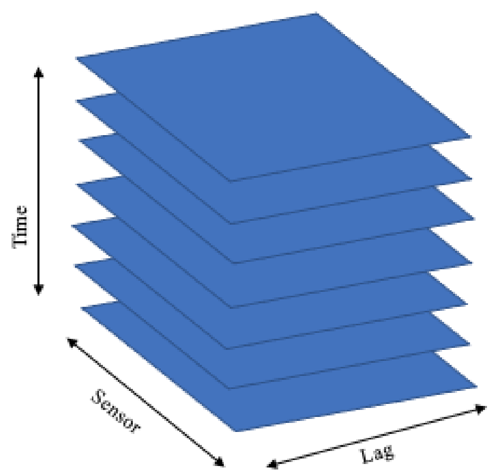

Figure 2: Representation of generated Tensor data

There are twenty-seven machines, and every machine has a similar set of the dataset which are shown below:

- Classes
- Classes 2
- Time id
- X range local
- Y(Alarm Types)
- Y2 (Normal and Alarm Type)
- Features
- Feature Id
- X raw

Each dataset contains different parameter lists. as describe below:

- Classes:This dataset contain list of different alarms types and Normal. The normal here represents no faults.
- Classes2: This dataset contains the two numeric data: 0 and 1, where 0 represents the normal and 1 represents the alarm(warning).
- Time id : It is time stamp from 1 to 720 per cycle. The 33 errors19 warnings maximum time stamp with 5 frequency time, it gives 1 hour (720x5=3600) of time period. Time id estimates a variable in the input data set for its suitability,as the time id variables are used for the Time series Analysis.
- X range local: This dataset is normalized data, which has features details in column and samples values in row.
- Y(Alarm Types) : This dataset provides details of different alarms(denoted as numeric digit of 1 to 20) used in the machines.
- Y2(Normal and Alarm Type) : This dataset contain 0 and 1 values which represents 0 as Normal and 1 as alarms
- Features: This dataset has list of features.
- Feature Id: It has feature Id corresponding to features list.
- X raw : It is raw data which has features in column and samples in row.Here in the thesis we are not using raw data,as these data is not normalized.

3.2 EXPLORATORY DATA ANALYSIS

In this section, we illustrate why preprocessing the time series dataset is necessary. We know that the point dataset is one in which data obtained is independent of time in which the data sets were obtained. We have some datasets where the time aspect plays a significant role in examining whether the points are normal or anomaly. It is one of the crucial and challenging parts of a project. Data quality assessment is required to clean the irrelevant, handle missing values, duplicates, data transformation, etc., to get the best accurate and quality results. Since we got the normalized data, however, when we have done the data assessments, we found that some of the machines have null values, so we have handled it by filling with median and removed the feature which had complete null values. For instance, below are the figures before and after handling from one of the machines.

1	2	3	4	5
-0.074	NaN	NaN	-0.54	-0.09
-0.074	NaN	NaN	-0.58	-0.098
-0.074	NaN	NaN	-0.58	NaN
-0.074	NaN	NaN	NaN	-0.078
-0.052	NaN	NaN	NaN	-0.095
-0.052	-0.052	NaN	-0.56	NaN

Table 1: Missing values before data cleaning

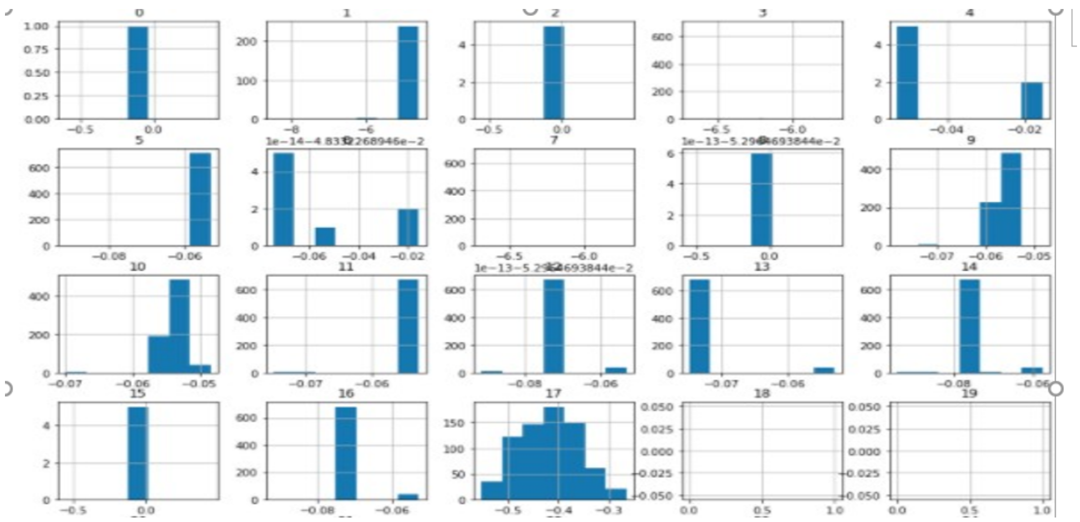

Figure 3: Visualisation of missing values before cleaning

1	2	3	4	5
ts_0	-0.07	-0.09	-0.07	-0.56
ts_1	-0.07	-0.09	-0.07	-0.57
ts_2	-0.07	-0.78	-0.07	-0.51
ts_3	-0.05	-0.78	-0.07	-0.54
ts_4	-0.05	-0.78	-0.07	-0.53

Table 2: Missing values after data cleaning

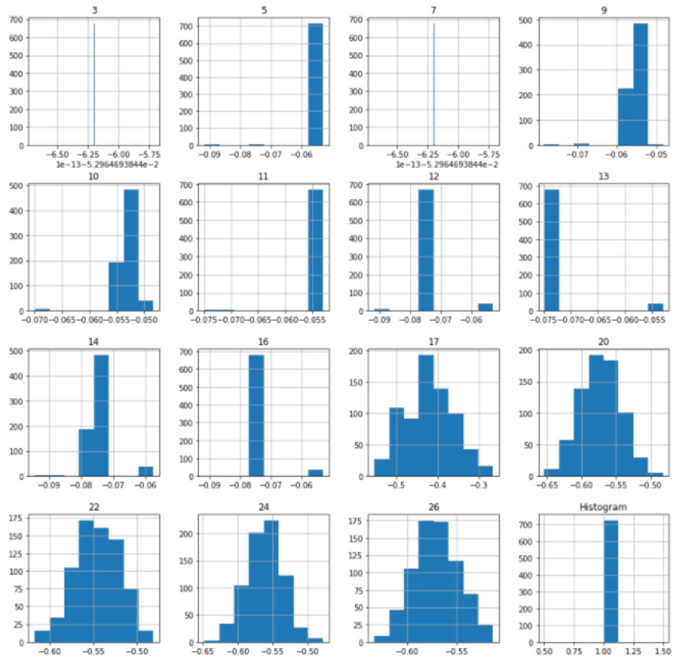

Figure 4: Visualisation of Missing values after data cleaning

3.3 ARCHITECTURE

In this thesis project, We have 27 machines; each machine has a set of different datasets. While experimenting, we have checked all machines and found the best machines among them based on several alarms samples present in the datasets. In some machines, several alarms samples are significantly less, so those machines are ignored for further experiments. After data collection, the datasets is split into train and test and then trained in the training set. In the Exploratory data analysis stage, data analysis and cleaning were done. Since we got the normalized datasets, they are filtered, however, after data analysis, it was seen that some features have null values, which were handled by taking the mean, or if the features have complete null values, then it was removed. After that, in the next stage "Dividing the windows", splitting the training set equally into sub-windows, moreover, before this step, we downsized the resolution of time series data by registering the mean of two sequential values then dividing the windows for modeling and further experiments.

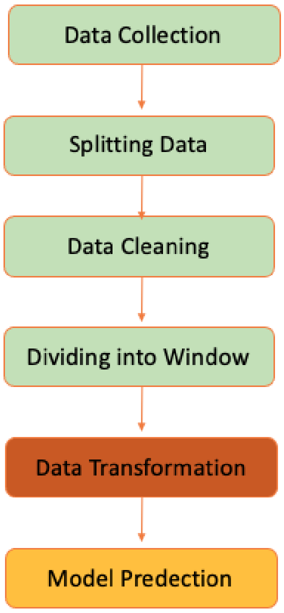

Figure 5: Framework for Proposed Method

In the next stage "Data Transformation", the preprocessed data was transform in different methods such as Vector, Matrix, and Tensor Regression. The Hankelization method is introduced in Matrix and Tensor Regression to see the effect. Also, another dimension lag added in Tensor Regression to check the performance and compare with Vector and Matrix methods.

For training model and prediction, we have used Regression, Neural Network models. In Neural Networks, Convolutional Neural Networks and LSTM regression models are used. As we have a time-series dataset, and it has been studied by the other researchers previously that neural network models perform better than the other models. Since, In our model experiments, we use Tensor with Hankelization method with additional dimensionality lag, which has not been experimented before in a similar scenario, so it is new experiment research to check the performance and compare with other methods.

We also have some additional research, checking the window time, which has data of multiple sensors and the different time instances. Evaluating the different windows with varying sizes, time granularity, and lag (for tensor only), moving the window ahead toward the alarm occurrence or caution event.The primary philosophy is to check with various settings (window size, time granularity lag) and find the optimized output results for each alarm in different machines with different models. The only difference in matrix and tensor is the

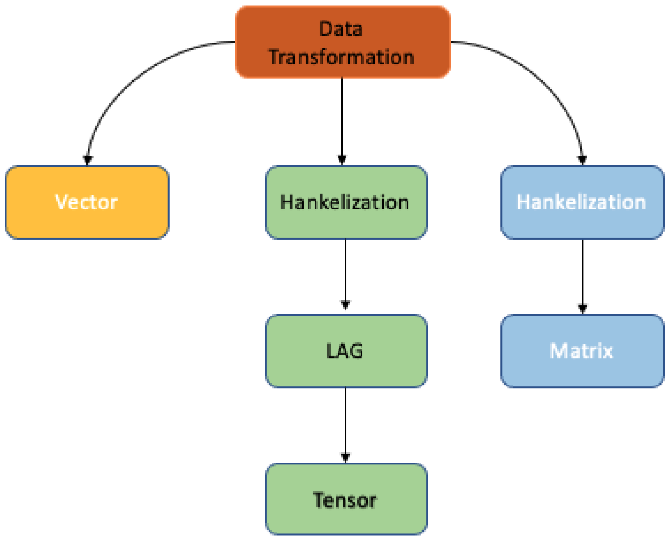

Figure 6: Transformation of preprocessed data

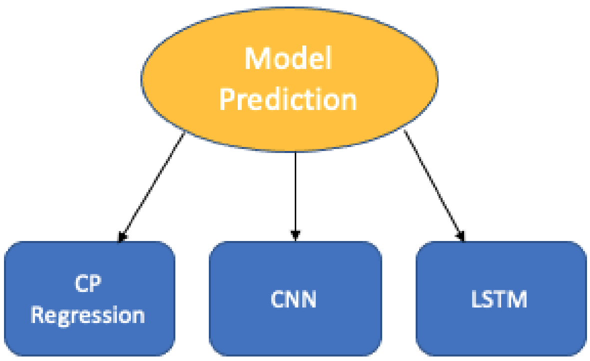

Figure 7: Prediction of distinct models

lag dimensionality, where we hankelized matrix with a lag to form tensor. So, there are four tensor measurements: time granularity, window size, samples, and lag. As experimental research, we are checking the outcomes on Vector, Matrix with Hankelization and Tensor with Hankelization, and lag with cp regressor. Neural network (CNN and LSTM) and trying to hyper tune the output by changing different architecture parameters matrices and predicting how long later the alarm is going to happen. In the previous study and research, it is observed that Tensor Regression performs better than Vector and

Matrix methods. So it is a hypothesis that the Tensor method performs better results in our experiments as well and also checking the effect of another dimensionality lag and Hankelization in the Tensor method, which is also an interesting experiment is observing in our experiment results. Vector is a matrix with just one row or column. A vector is a one-dimensional or first-order tensor.Matrix is a two-dimensional or second-order tensor. It's anything but a matrix of n Ã m numbers encompassed by sections. We can add and deduct matrices of a similar size, duplicate one lattice with another as long as the dimensions are viable, and multiply a whole network by a constant.A matrix is only a 2-D lattice of numbers.

.

3.4 HANKELIZATION

A matrix is called a Hankel matrix, If all its elements along the skew-diagonals are constant. The process of generating a Hankel grid/tensor from a given information vector/matrix/tensor is called as Hankelization [18]. It is a tensorization techniques that makes uses of hankel matrix. Hankel matrix is useful for signal decomposition and temporal frequency representation. The main objective of hankelization is to obtain high order tensor matrix.

$$\mathbf{X}_H = \mathcal{H}_\tau(\mathbf{X}) := \begin{pmatrix} x_1 & x_2 & \cdots & x_{N-\tau+1} \\ x_2 & x_3 & \cdots & x_{N-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_\tau & x_{\tau+1} & \cdots & x_N \end{pmatrix} \in \mathbb{R}^{\tau \times (N-\tau+1)}.$$

Figure 8: Hankelize matrix

The hankel matrix of XH is given as follows

3.5 CONVOLUTIONAL NEURAL NETWORK

CNN's are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Deep Learning algorithm can take in an input image, assign importance (learnable weights and biases) to various aspects in the image, and differentiate one from the other[16]. It is also interesting to check with CNN as it is widely used to analyze visual images and relate with our experiment methods and Consider as a four-dimension image. For the hyper tunning, we experimented with different architecture parameters to optimize the results by checking the result with various activation functions, different optimizers, increasing The epoch size, adding filter layers, dense layer, and dropout layer.

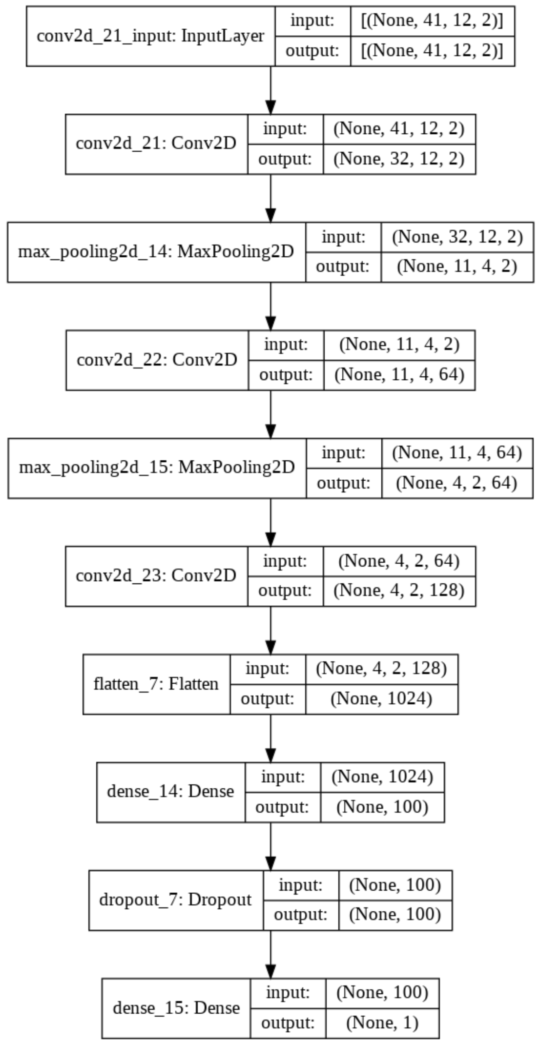

Figure 9: Plot model of CNN Architecture

Activation function	ReLU
Number of layers	3
Layer Type	Conv2D
Epochs	100
Optimizers	Adam
Cost Function	Mean Absolute Error
Drop out	0.5
Number of filters	32,64,128
Filters size	3x3

Figure 10: Model Parameters of CNN Network

3.6 LONG SHORT TERM MEMORY

LSTM networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. It is widely used in time series forecasts. It rectifies huge issues that recurrent neural networks suffer from short-memory. Using a series of gates, each with its own, the LSTM manages to keep, forget or ignore data points based on a probabilistic model[17]. For hyper tuning, In this also, we have experimented with different architecture parameters to optimize the results, such as by checking the result with various activation functions, other optimizers, increasing the epoch size, adding filter layers, dense layer, dropout layers.

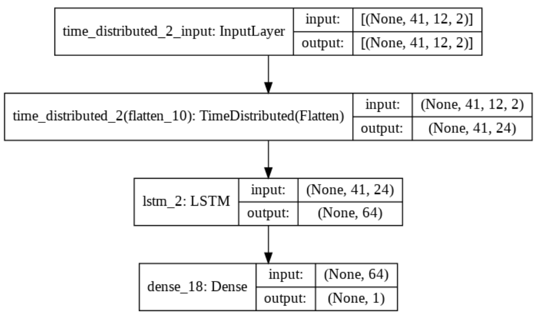

Figure 11: Plot Model of LSTM Architecture [14]

Activation function	ReLU
Number of layers	2
Epochs	50
Optimizers	Adamax
Cost Function	Mean Absolute Error
Drop out	0.3
Filters	64,128

Figure 12: Model Parameter of LSTM Network

3.7 PREDICTIVE MAINTENANCE

Predictive maintenance is a proactive maintenance strategy that minimizes a systems downtime or failure by predicting failures before they happen. It uses data from sensors to measure the components state of health and find patterns that can help to make forecasts about its future degradation [9]

his approach has lots of benefits for an organization. It helps to avoid maximum uses of machine components, detect early anomaly patterns and provide statistics that allows engineers to know when maintenance or repairs should be taken. If the system is not taken care of at the right stage or measured too late, the cost of failure and production downtime would be huge for an organization. Using predictive maintenance with machine learning can be easy to adjust new data dynamically, measure the real-time behavior of the system, and predict future maintenance plans. It does not require manual configuration, threshold setting, which another maintenance Strategy requires.

3.8 REMAINING USEFUL LIFE

The remaining Useful Life regression approach tells how much time is left before the failure. It is the length of time when a machine is expected to operate before it requires maintenance or replacement in advance. By considering Remaining useful life, engineers can plan the maintenance, avoid unplanned downtime, optimize performance efficiency; therefore, the remaining functional approach is a top priority in predictive maintenance strategy[16] It not only predicts the remaining useful life but also gives confidence in the prediction. The Remaining useful life is described from the below figure[13].

The whole process is divided into four stages, i.e., Health, Caution, Repair, and Failure. The health index ranges from 0 to 1; it shows the health status of a system, and the more the range is, the healthier the system. The values are determined based on the condition of a system. The time t1-to gives the Remaining useful life system estimation. When the system is dropping it as healthier level, it shows the caution to be aware that the system needs maintenance. This time to-t1 is when the system started deterioration till the endpoint of deterioration, which is last acceptable health and repairing is done. After the endpoint of an acceptable health level, the system goes to a failure state. While predicting Remaining useful life, We are using Root Mean Squared Error since it is used to measure the difference between actual and predicted values by estimator and penalizes significant errors severely, and that would force the algorithm to forecast Remaining useful life as close as possible, so we achieve to reduce the

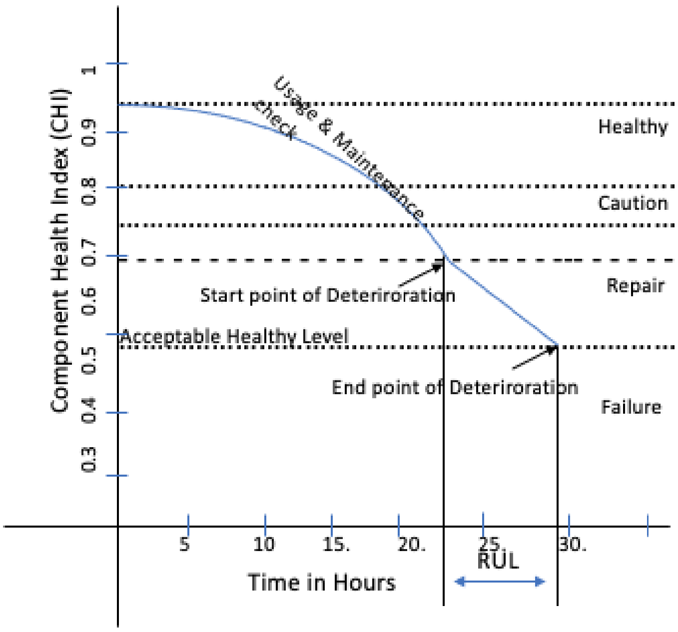

Figure 13: Remaining useful life [16]

error between actual Remaining useful life and the predicted Remaining useful life.

3.9 DATA SEGMENTATION

Data segmentation process involves taking the complete data and splitting it into equal window size. The window size represents the amount of data to be presented, and the overlap size represents the number of repeated data from the previous window frame.

In the time dataset, it has a cycle of 1-to-720-time id, which is an hour complete-time window (720 time instant and 5 sec per sample frequency-time = 3600 seconds) and then dividing the complete window into small windows (sample size * per sample frequency-time) by taking the average of it, so, the complete window would be split into the exact time frame and based on input window, it will calculate the target (total time window (minus) input time window). For example, If the first window has 30 samples and 5 sec per sample frequency time, then the first window will be 150 seconds (30x5 =150 sec). The complete input window will be dividing into 24 small subwindows (150*24=3600), then based on the first window, the target will be calculated as (720x5-30x5= 3450 sec), it can be seen in fig[15], then, we will predict after how long time later the alarm will happen.

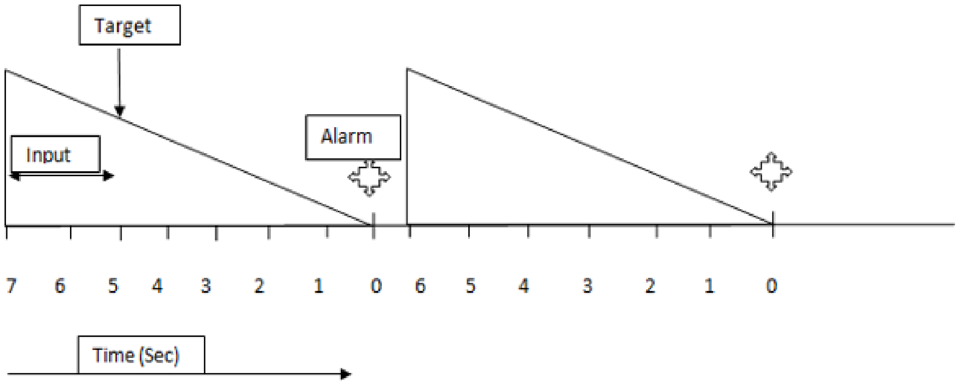

Figure 14: Time to next alarm event
[13]

After the first window, it will move the window ahead towards the alarm and calculate the target and predict the alarm to happen; it can be seen in fig[16]. Similarly, the window will slide towards the alarm and calculate until the last windows shown in fig[17].

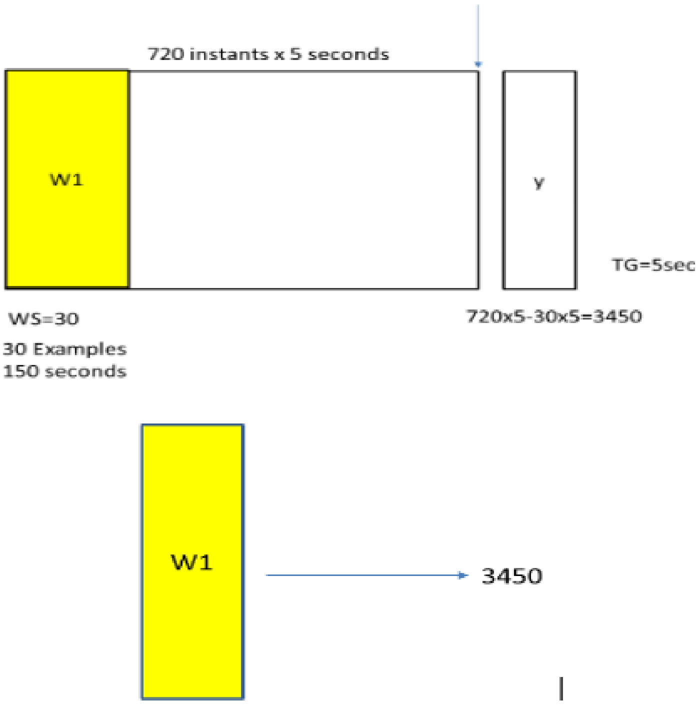

Figure 15: Segmentation of Window for first event

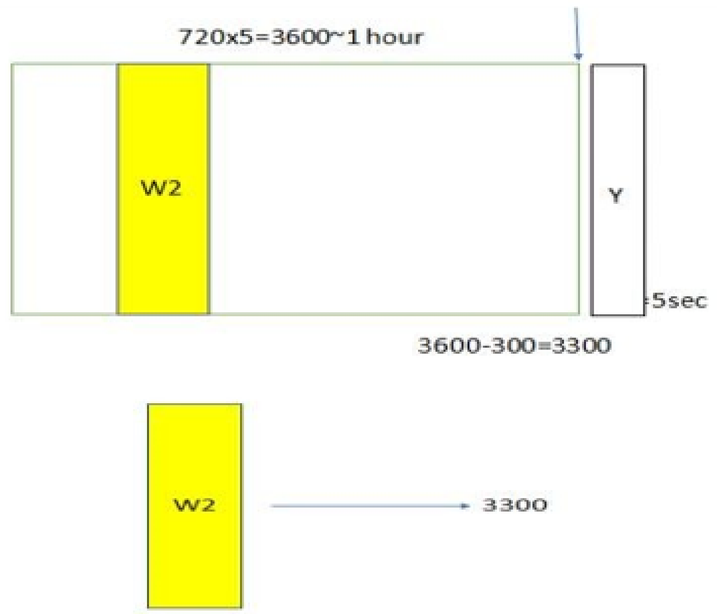

Figure 16: Segmentation of window for second event

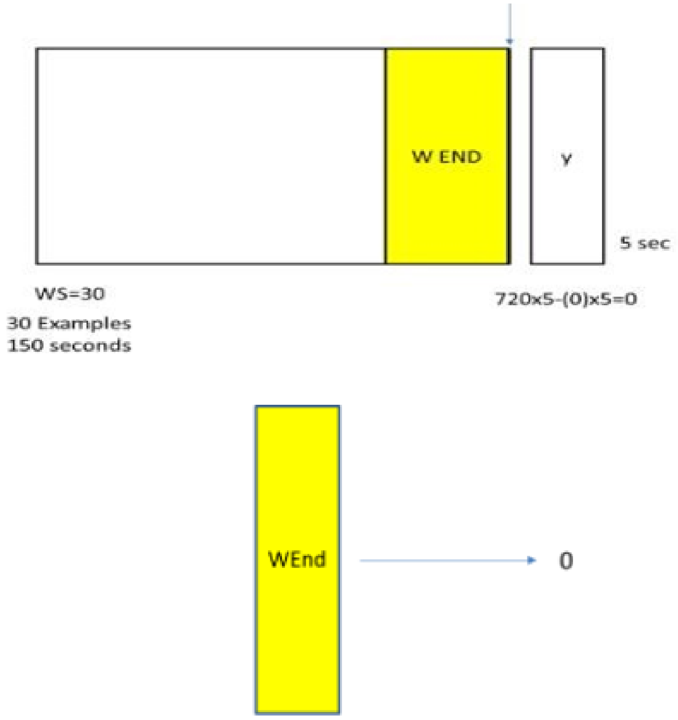

Figure 17: Segmentation of window for final event

4 RESULTS

In this result section, we checked the performance of all the alarms present in machines 13,14 and 15. The figure[18]and[19] illustrates the output of alarm A69 Train and test mean absolute error for Machine 13 and 14. The train and test mean absolute error is in seconds. We have checked the results with different configuration settings such as sampling frequency, window size, lag dimension and picked the best performed output among them, i.e., low error for different algorithms.

In our evaluation, we have checked with Tensor Regression, Tensor - Convolutional neural network, Tensor - Long short-term memory and compared with Vector linear regression and Matrix regression. The architecture for tensor and matrix regression is slightly different from than vector method. We used Hankelization in Tensor and Matrix Regression. In Tensor addition dimension also added that is lag. As we could see from the figure that the tensor regression has a better performance compared to the other methods. The train mean absolute error is less for tensor regression; however, the variance between the train and the test is more, which shows overfitting, and the potential reason would be the non-linearity of the dataset.

Alarm A69	Machine 13		Machine 14	
Methods	Configuration	Train MAE	Configuration	Train MAE
Tensor Regression	(TG:5,W:10,L:2,S1812)	633,9	(TG:20,W:5,L:8,S546)	602,8
Tensor-Convolutional Neural Network	(TG:20,W:5,L:8,S906)	808,9	(TG:10,W:10,L:4,S546)	808,9
Vector linear Regression	(TG:20,W:5,S906)	880,3	(TG:15,W:1,S3640)	1057,6
Matrix Regression	(TG:15,W:5,S1208)	704,4	(TG:15,W:5,S1208)	704,2
Tensor - lstm	(TG:15,W:2,L6,S330)	834,8	(TG:5,W:2,L2,S5460)	885,4

Figure 18: Alarm A69 Train MAE Results for Machine 13 and 14

Alarm A69	Machine 13		Machine 14	
Methods	Configuration	Test MAE	Configuration	Test MAE
Tensor Regression	(TG:5,W:10,L:2,S1812)	1080,8	(TG:20,W:5,L:8,S546)	880,8
Tensor-Convolutional Neural Network	(TG:20,W:5,L:8,S906)	831,1	(TG:10,W:10,L:4,S546)	829,1
Vector linear Regression	(TG:20,W:5,S906)	1242,3	(TG:15,W:1,S3640)	1276,2
Matrix Regression	(TG:15,W:5,S1208)	1124,6	(TG:15,W:5,S1208)	919
Tensor - lstm	(TG:15,W:2,L6,S330)	972,8	(TG:5,W:2,L2,S5460)	923,4

Figure 19: Alarm A69 Test MAE Results for Machine 13 and 14

While looking into the output result of the Tensor convolution neural network, we could say that variance in the train and test mean absolute error is less. The test result outperformed the other methods. Different architecture parameters are used in this method, such as epochs, activation layer, and dropout layer, to improve the model's performance. On the contrary, when we see the result of vector linear regression, we can say that the mean absolute error for both cases are relatively high compared to the other methods. This applies the same in the case of Tensor Long short term memory (LSTM)and Matrix regression as the results are not much optimal compared to the tensor methods. We know that the LSTM is capable of learning order dependency in sequence prediction problems. Still, in our outcome, the performance of LSTM is not optimal. Thus according to the above discussion, we can say that the tensor regression method performs better than other methods

.

While examining, we take Machine 13 which contains three alarms, namely A33, A69, and A502. We compare the results of each alarm to check how well it performs on different methods. The Alarm A33 works better with the tensor method with the time granularity of five seconds, a window size of ten minutes with a lag of 2.

Further ahead, we checked the performance of different alarms, compare the results with other machines, and validate which configuration gives the optimal results. The figure[20] represents the results of alarm A502 train mean absolute error for machine 13 and machine 15 respectively. With the time granularity of 5 seconds, window size of 2 minutes with the lag of 2, we could say that the results obtained for tensor regression in machine 15 is better than the machine 13, as it has a better mean absolute error for train case. The only difference is that the configuration varies for each alarm.

Alarm A502	Machine 13		Machine 15	
Methods	Configuration	Train MAE	Configuration	Train MAE
Tensor Regression	(TG:5,W:2,L:2,S330)	580,4	(TG:5,W:2,L:2,S360)	480,4
Tensor-Convolutional Neural Network	(TG:10,W:1,L:2,S660)	824,2	(TG:5,W:1,L:2,S720)	827,2
Vector linear Regression	(TG:5,W:3,S220)	974,9	(TG:5,W:2,S360)	774,6
Matrix Regression	(TG:5,W:2,S330)	584,5	(TG:15,W:1,S240)	589,1
Tensor - lstm	(TG:10,W:1,L4,S330)	834,8	(TG:5,W:1,L2,S720)	969,4

Figure 20: Alarm A502 Train MAE Results for Machine 13 and 15

Similarly, the figure[21] represents the results of alarm 502 test mean absolute error for machine 13 and machine 15. The test mean absolute error in machine 15 is better than machine 13, however, when compare to the non- linear model like Tensor convolutional Neural Network the test mean absolute error is better than the tensor network ,this is because of hyper tunning of tensor convolutional neural network which robust to overfitting, and has less variance in the train and test result.

Likewise, we are checking for different machines to evaluate how the alarms perform for other methods. The figure[22],[23] illustrates the performance of alarm A33 train mean absolute error for Machine 13,14 and 15. We can see that the alarm results for machine 13,14 and 15 are similar performance result, with some variances of output result values such as the train mean absolute error in machine 15 is

Alarm A502	Machine 13		Machine 15	
Methods	Configuration	Test MAE	Configuration	Test MAE
Tensor Regression	(TG:5,W:2,L:2,S330)	1013,5	(TG:5,W:2,L:2,S360)	991,6
Tensor-Convolutional Neural Network	(TG:10,W:1,L:2,S660)	935,3	(TG:5,W:1,L:2,S720)	980,9
Vector linear Regression	(TG:5,W:3,S220)	1190,1	(TG:5,W:2,S360)	1150,1
Matrix Regression	(TG:5,W:2,S330)	1137,2	(TG:15,W:1,S240)	1177,3
Tensor - lstm	(TG:10,W:1,L4,S330)	972,8	(TG:5,W:1,L2,S720)	1252

Figure 21: Alarm A502 Test MAE Results for Machine 13 and 15

better than machine 13 and machine 14.The time granularity of 20 seconds, and window size of 5 minutes with lag 2 works better in machine 15, when compared to the other machines.

Alarm A33	Machine 13		Machine 14	
Methods	Configuration	Train MAE	Configuration	Train MAE
Tensor Regression	(TG:5,W:10,L:2,S1806)	624,4	(TG:20,W:5,L:8,S543)	688,2
Tensor-Convolutional Neural Network	(TG:20,W:5,L:8,S903)	816,9	(TG:20,W:5,L:8,S903)	867
Vector linear Regression	(TG:5,W:5,S3612)	980,3	(TG:5,W:5,S2172)	926,5
Matrix Regression	(TG:5,W:10,S1806)	721,4	(TG:5,W:10,S1086)	776,8
Tensor - lstm	(TG:10,W:1,L4,S9030)	894,2	(TG:15,W:10,L6,S362)	903,1

Figure 22: Alarm A33 Train MAE Results for Machine 13,14

Alarm A33	Machine 15	
Methods	Configuration	Train MAE
Tensor Regression	(TG:20,W:5,L:8,W897)	611,5
Tensor-Convolutional Neural Network	(TG:15,W:10,L:8,S598)	562,3
Vector linear Regression	(TG:5,W:1,S17940)	883,9
Matrix Regression	(TG:15,W:10,S598)	664,3
Tensor - lstm	(TG:15,W:2,L6,S1810)	949,8

Figure 23: Alarm A33 Train MAE Results for Machine15

Alarm A33	Machine 13		Machine 14	
Methods	Configuration	Test MAE	Configuration	Test MAE
Tensor Regression	(TG:5,W:10,L:2,S1806)	883,6	(TG:20,W:5,L:8,S543)	923,6
Tensor-Convolutional Neural Network	(TG:20,W:5,L:8,S903)	843,2	(TG:20,W:5,L:8,S903)	835,3
Vector linear Regression	(TG:5,W:5,S3612)	1104,8	(TG:5,W:5,S2172)	1193,8
Matrix Regression	(TG:5,W:10,S1806)	1034,1	(TG:5,W:10,S1086)	1080
Tensor - lstm	(TG:10,W:1,L4,S9030)	920,4	(TG:15,W:10,L6,S362)	965,7

Figure 24: Alarm A33 Test MAE Results for 13 and 14 machines

Alarm A33	Machine 15	
Methods	Configuration	Train MAE
Tensor Regression	(TG:20,W:5,L:8,W897)	611,5
Tensor-Convolutional Neural Network	(TG:15,W:10,L:8,S598)	562,3
Vector linear Regression	(TG:5,W:1,S17940)	883,9
Matrix Regression	(TG:15,W:10,S598)	664,3
Tensor - lstm	(TG:15,W:2,L6,S1810)	949,8

Figure 25: Alarm A33 Test MAE Results for Machine 15

The figure[24],[25] illustrates the results of alarms A33 test mean absolute error for machine 13,14,15 .The mean absolute error in tensor convolution neural network has a slight edge over the other methods.Though we observed in figure[22],[23] the tensor representation was better in train cases, but due to over fitting, the tensor convolutional neural network has a better performance than the tensor network in terms of low variance of train and test results. In our preliminary work, we selected the machines out of 27 machines based on more number of data samples and then used datasets from three machines, namely 13, 14, and 15. In the preprocess data, applied cross validation to prevent over fitting. We have verified with different cross validation values and compared the model performs. The cross-validation score of ten performs reasonably better than the other scores

.

Hence, from the above evaluation results, we could conclude that the performance of Tensor regression in all the alarms present in the selected machines outperforms the other methods and Tensor Convolutaional neural network gave the best result in terms of low variance.

5 CONCLUSION

In our thesis, we proposed Tensor Regression with Hankelization and added additional dimension lag to check the performance on multi-sensor unequal-length time series machine datasets, then compare the performance results with the Matrix Regression method, which does not have the additional dimension lag and Vector linear regression method which is linear regression in vector form without Hankelization and lag dimension.

With this experimental research-based project, we also found the ideal configuration settings such as ideal window size for taking samples before the occurrence of alarms and warnings, ideal sampling frequency (time granularity) for making better predictions, best lag dimensional value. This set of configuration settings are one of the major factors for the optimal result and a better predictive model. It is also observed during the experiments that lower the time granularity(Sampling frequency), the higher the variance on the result in most of the cases.

We have twenty-seven machines, and the initial step towards the goal was, finding the best machine for training and testing purposes. After verifying, we secured a few machines(13, 14 and 15) based on the number of samples are present for a particular alarm feature. After evaluating on selected machines, we can conclude that the model output performance is approximately similar in the machine 13,14, and 15. Tensor Regression model performs best results than other models on train set and potential reason would be due to Hankelization technique and lag affect which is not present in Vector and Matrix Regression, however, could see the overfitting on Train and Test in Tensor Regression output result, it could be because of non-linearity of data or Covariate shift. Moreover, Tensor convolutional neural network is robust to overfitting and also performed well in terms of low variance in Train and Test output. A vector linear regression model is a a parametric algorithm, which is not flexible to learn the complex dataset, on another side, Tensor Regression is a multi-dimension and capable of handling complex structure data and the Hankelization with lag dimension influences the performance of the model. Matrix regression is a slightly lower performance output than the Tensor regression, and it may be due to missing lag dimension in the architecture. The tensor LSTM model performed better than the Vector Regression, but less comparable to Tensor Regression and Tensor Convolutional neural network.

Thus this proposed model is not only better than the existing alarm functioning based on predefined threshold limit system but also provide the smart alarm system that help the company to predict the upcoming failure in the machines and improve the machines useful life, avoid downtime, enhance the efficiency of production quantity and qualities and prevent damages or fault.

Our goal for future work is to reduce the variance in tensor regression.To reduce the non-linearity effect,we can create parallel models for different types of alarms or implement different models for different groups of signals.

The claim is to demonstrate which group of models best matches for our data set. For an instance to verify whether it is linear, multi-Linear or non -linear. We prefer multi- linear model(tensor regression) in our case as we can see the performance is better than the other methods.

BIBLIOGRAPHY

[1] Taufik Nur Adi, Hyerim Bae, and Nur Ahmad Wahid. Automatic false-alarm labeling for sensor data. *Journal of The Korea Society of Computer and Information*, 24(2):139–147, 2019.

[2] Maame G Asante-Mensah, Salman Ahmadi-Asl, and Andrzej Cichocki. Matrix and tensor completion using tensor ring decomposition with sparse representation. *Machine Learning: Science and Technology*, 2(3):035008, 2021.

[3] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019.

[4] Adriana Horelu, Catalin Leordeanu, Elena Apostol, Dan Huru, Mariana Mocanu, and Valentin Cristea. Forecasting techniques for time series from sensor data. In *2015 17th international symposium on symbolic and numeric algorithms for scientific computing (SYNASC)*, pages 261–264. IEEE, 2015.

[5] Ming Hou. Tensor-based regression models and applications. 2017.

[6] WH Wan Ishak, Ku-Ruhana Ku-Mahamud, and Norita Md Norwawi. Mining temporal reservoir data using sliding window technique. *CiiT International Journal of Data Mining Knowledge Engineering*, 3(8):473–478, 2011.

[7] Pooja Kamat and Rekha Sugandhi. Anomaly detection for predictive maintenance in industry 4.0-a survey. In *E3S Web of Conferences*, volume 170, page 02007. EDP Sciences, 2020.

[8] Joel Levitt. *Complete guide to preventive and predictive maintenance*. Industrial Press Inc., 2003.

[9] R Keith Mobley. *An introduction to predictive maintenance*. Elsevier, 2002.

[10] Elizabeth Newman, Lior Horesh, Haim Avron, and Misha Kilmer. Stable tensor neural networks for rapid deep learning. *arXiv preprint arXiv:1811.06569*, 2018.

[11] Gregory C Reinsel. *Elements of multivariate time series analysis*. Springer Science & Business Media, 2003.

[12] Tomáš Šabata, Juraj Eduard Páll, and Martin Holena. Deep bayesian semi-supervised active learning for sequence labelling.

In *Workshop & Tutorial on Interactive Adaptive Learning*, page 80, 2019.

[13] Hanan A Saeed, Hang Wang, Minjun Peng, Anwar Hussain, and Amjad Nawaz. Online fault monitoring based on deep neural network & sliding window technique. *Progress in Nuclear Energy*, 121:103236, 2020.

[14] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[15] Bernard Schmidt, Ulf Sandberg, and Lihui Wang. Next generation condition based predictive maintenance. In *The 6th International Swedish Production Symposium 2014 16-18 September 2014*, 2014.

[16] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14, 2011.

[17] Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019.

.

HALMSTAD
UNIVERSITY

PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se