**Paper ID: 585**

# A **Gated Recurrent Unit Neural Network based Predictive Maintenance Approach for Machinery Maintenance in the Apparel Industry**

UMMPK Nawarathne[1#], Chamila Walgampaya[2], and Iroshan Aberathne[3]

[1] *Computing Centre, Faculty of Engineering, University of Peradeniya, Sri Lanka*
[2] *Department of Engineering Mathematics, Faculty of Engineering, University of Peradeniya, Sri Lanka*
[3]*Faculty of Agriculture & Life Sciences, Lincoln University, New Zealand*
[#]mnawarathne20@gmail.com

***Abstract****—The Sri Lankan garment industry has been garnering attention by bringing the country a huge income over the past years. The role of well-functioning machinery is a crucial factor in producing flawless products in this industry. Hence it is a must for machinery equipment to work regularly thereby providing the engineering crew a minimum hassle. Therefore this research paper presents a predictive maintenance (PdM) based methodology designed with the aid of a type of deep learning model, a Gated Recurrent Unit Neural Network (GRU) to predict a machinery breakdown due to component failures. Machinery data were used to create data models which gave the component malfunctioning as a multiclass classification output. While researching, to handle the class imbalance problem, Synthetic Minority Oversampling Technique (SMOTE) mechanism was also used to obtain a balanced data distribution. Various combinations of basic deep learning models and models based on Recurrent Neural networks (RNN), GRU, and Long Short-Term Memory networks (LSTM) were used to train the data models, where the GRU-SMOTE model outperformed the other models that had an accuracy of 98.77% along with fine scores for macro average precision, macro average recall, and macro average f1-score. These early hand predictions can be therefore utilized to face sudden machinery failures that will allow the mechanical crews to plan and schedule maintenance work efficiently preventing the expenditure of unnecessary time and resource wastages.*

***Keywords****— deep learning, multi class classification, predictive maintenance, gated recurrent unit networks*

## I. Introduction

The modern world has been adopting artificial intelligence (AI) and machine learning (ML) technology based approaches deeply almost in every field due to the vast advantages found in the usage (Wuest et al., 2016). The apparel industry also has been driven toward the usage of these polished methods. The Sri Lankan apparel industry which has been operating since the 1980s has provided so many advantages to the country's income and as well as given the Sri Lankan community with so many employments (Dheerasinghe, 2009). With the surge of popularity in the production of garments, this industry also has faced various threats that could hinder garment production which has made the use of AI-based methods to be utilized as solutions.

The good performance of machinery and pieces of equipment in garment production is a crucial factor that has to be considered during the manufacturing process. Even the malfunctioning of a single machinery could cause the whole process to break down causing a vast profit loss to the garment factory. The common methods to handle such unexpected mishaps during most field operations were using remedies such as Run-to-Failure (R2F) which is the simplest method to handle the issue by treating at the occurrence time and Preventive Maintenance (PvM) another methodology that is used to schedule maintenance measures beforehand that is better than the previous method (Guduru et al., 2018). However, both these methods have been identified to be very costly in terms of resource and time wastage that specially made the researchers explore alternative approaches (Susto et al., 2015).

Predictive Maintenance (PdM) is one such approach that is currently being preferred widely due to the ability to provide advanced detection of pending failures and enable timely pre-failure interventions of future predictions (Susto et al., 2015). This technique is considered better than other methodologies that are Run towards Failure (R2F), and Preventive Maintenance (PvM) due to the resources and time cost saving ability (Susto et al., 2015). The usage of PdM has hence provided the end users with past behavioral records to study the history of the assets and thereby allowing prediction of happenings based on the data. The above mentioned PvM and PdM methodologies are built using the basis of ML techniques as described in (Guduru et al., 2018) and (Susto et al., 2015).
Evolving from the traditional approaches such as the R2F method, the manufacturing world now has arrived at using advanced technologies that use machine

learning and artificial intelligence due to possessing abilities like being able to handle high dimensional and large datasets, and analyze patterns using data (Wuest et al., 2016). Noting the ability of ML, to learn and adapt to changing environments using data records, the authors of the paper (Wuest et al., 2016) have described various ML algorithms and methods used by researchers in the manufacturing area to provide solutions. This paperwork has hence summarized the importance and application of machine learning techniques in the area of manufacturing.

## II. Literature Review

The authors (Guduru et al., 2018) discussed the usage of PvM with the usage of genetic algorithms in implementing the technology to assist in the operation of sewing machines used in garment industries. The introduced implementation has been conducted using dynamic programming where mathematical equations have been derived related to machinery breakdown costs. Even though it has been proven that PvM has reduced the costs in this use case, the work presented by the authors in (Wuest et al., 2016) has explained how PdM has become a better technology than PvM showing the importance of the former practice.

The research work presented by (Susto et al., 2015) contains a detailed description of the importance of the PdM methodology over other techniques that are used to tend to machinery related failures. The authors have introduced a novel idea called the ‘Multiple Classifier’ (MC) PdM methodology while justifying the reasons for using PdM in their work. This research work has emphasized the importance of PdM thoroughly.

In order to forecast potential failures and quality flaws, A. Kanawaday and A. Sane (Kanawaday and Sane, 2017) investigated the use of AutoRegressive Integrated Moving Average (ARIMA) forecasting on time series data collected from various sensors from a slitting machine that will enhance the overall manufacturing process. The primary goal has been to provide industrial machines with prognostics in order to boost productivity and prevent quality failures.

A. Wahid, J.G. Breslin, and M.A. Intizar (Wahid, Breslin and Intizar, 2022) have used Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) with skip connection as a hybrid model (CNN-LSTM) combined in the method for multivariate time-series forecasting for predictive maintenance (PdM). To predict machine failures, the authors tested the CNN, LSTM, and CNN-LSTM forecasting models one at a time on the same data that was also used in this research, producing a model accuracy of 96.8% for the best model which was the CNN-LSTM model, while using oversampling methods to handle the data imbalance issue.

However, very less amount of studies has been done that use RNN-based technologies to predict machine failures. Our research, therefore, has been conducted by testing models between basic deep learning models, RNN models, GRU models, and LSTM models. As the RNN based models, have capacities to handle sequential data, capture long-term dependencies, handle variable-length sequences, and extract meaningful features, these model types are well suited for predictive maintenance tasks.

In this paper therefore, a deep learning procedure has been followed to predict machinery failures ahead of time. Deep learning (DL) is an advanced version of the machine learning area (Serradilla et al., 2022). Hence, a type of recurrent neural network (RNN), a Gated Recurrent Unit Neural Network (GRU) has been used to predict the machinery malfunctioning. The GRU networks also have become popular in recent years due to their ability to memorize past data through lengthy windows thereby overcoming this shortcoming set by traditional feature engineering techniques (Dey and Salem, 2017). It is expected that these early warnings will allow the mechanical crew to tend to problems efficiently and as well as the records that are used to train this problem would help in planning operations also.

This paper is organized such that section III discusses the methodology with the experimental design, section IV outlines the results while section V covers the results and the section VI provides this study’s conclusion.

## III. Methodology and Experimental Design

### A. Data Acquisition

The dataset found in the repository (Patel, 2018) was used in this research work that had 876100 records which is also a renown Microsoft case study. The dataset contains details of hundred similar machines that constitute machinery details, machinery component failures, maintenance records, nonbreaking errors thrown by them, and machinery telemetry data recorded every hour per day. Figure 1 depicts the details of the complete dataset with the respective data types.

| Column Name | Data Type |
|---|---|
| datetime | Datetime64[ns] |
| machineID | int64 |
| volt | float64 |
| vibration | float64 |
| rotate | float64 |
| pressure | float64 |
| errors_count | int64 |
| replaced_component | object |
| failed_component | object |
| model | object |
| age | int64 |

Figure 1. Dataset details

According to Figure 1,

1. *Volt*, *pressure*, *rotate*, and *vibration* are machinery readings that were reported under telemetry data.

2. The *errors count* depicts the non severe errors count thrown by the machines that has five error types, error 1, error 2, error 3, error 4, and error 5.

3. The *replaced component* has been used to make note of component maintenance. In total, this dataset has four component types, namely component 1, component 2, component 3, and component 4.

4. The *failed component* provides component failures that was also considered to be the target column during the model training.

5. Columns of machine id, model, and age provide the details of each machine.

All the readings of telemetry data, errors count data, component maintenance, and component failures have been recorded with the reported date and time. With the availability of records, the problem was trained as a supervised machine learning problem that uses historical data to observe the machinery's behavior to predict component failures. The overall steps that were carried out in the study have been depicted in Figure 2 with a diagram.

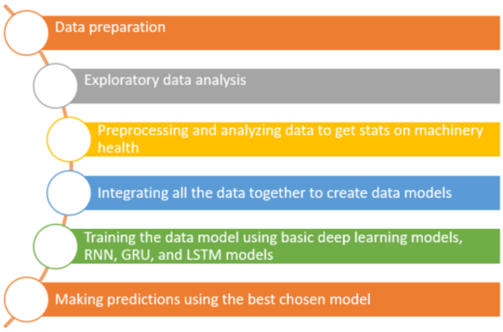



Figure 2. The overall workflow of the conducted research

Therefore as the first step, exploratory data analysis was held to understand the distribution of the dataset with the usage of Python libraries. Since the original dataset was clear of null and noisy values no records were removed during the preprocessing stage.

### B. Preparation of Data for Training Purposes

#### 1) *Maintenance Dataset*

To begin with integrating the datasets, first using the maintenance dataset, the machinery component replacement frequencies were calculated in an ad-hoc engineering method which is a very common application in predictive maintenance use cases (Patel, 2018). Hence the frequency of the component replacements was calculated which gave the output of how many times a component was replaced during maintenance.

#### 2) *Telemetry Dataset*

Using telemetry data with the time stamps, lag features were derived. Lag features are values from past time steps that can be useful since they are created under the presumption that the past can affect or might contain intrinsic knowledge about the future. Therefore from telemetry data, lag features such as rolling mean and standard deviation were calculated for voltage, pressure, vibration, and rotation readings considering every 6 hours in a lag window of 6 hours. To observe a long term time effect, a lag window of 24 hours was also used which gave insights related to the machinery's health. Hence from the telemetry data, the rolling mean and rolling standard deviation hours were calculated. It should be noted that while calculating statistics related to the machinery health, this research considered statistics per every 6 hours during lag windows of 6 hours and 24 hours after testing for other hour combinations. Therefore, these time values can be changed, and as well as other measures such as maximum, and minimum values can be used to describe machinery health accordingly.

#### 3) *Errors Dataset*

In a similar method to calculating the lag features of telemetry data, the frequency of errors that were been thrown also was calculated considering lag features due to the presence of timestamps in data. However as error ids were present in the data which is a categorical variable, error data were not averaged over time. Alternatively, error counts thrown per each error id during a lagging window were derived. Following steps 1 and 2, the data at this point was comprised of component replacement frequencies, frequency of thrown errors, and telemetry related calculated data.

#### 4) *Integrating with Failures and Machinery Data*

Finally, the failures related data and machinery data were merged into the prepared dataframe following steps 1, 2, and 3 that finished the data integration task. The dataframe at this step had all five datasets integrated together.

#### 5) *Final Dataframe Preparation*

The categorical variables such as the model type, replaced component type, and failed component type of the final dataframe were then converted to numerical values using dummy variables. Null values were removed to obtain an error-free dataframe for the model training purpose in the final dataframe. Finally, the components failures were prepared as the target column.

### *C.* Usage of Multi-class Classification and Model Training using Deep Learning Techniques

To train this problem, instead of the traditional train and test data splitting, a time window approach was followed to split data. Therefore, after testing with several time windows, data till the month of August were used in the training dataset while the rest of the data from August onwards were used for the testing dataset since this combination gave optimal results during the training phase. This also prevented from data being shuffled while preserving the timely order. After performing all the required data transformations, the data were trained using several deep learning models. Deep learning is an advanced machine learning technique that is comprised of a neural network having several layers that uses Graphics Processing Units (GPUs) to work efficiently (Serradilla et al., 2022). Since the outcome was to predict a certain component failure or a non-failure out of the four components, using multi-class classification concepts *categorical_crossentropy* was used as the loss function while various optimizers were tested for selection while training the model. The Keras library was used to perform the model training and predictions (Chollet et al., 2015). The data model was trained by testing using various combinations of neural network architectures while fine-tuning the respective model.

### *D.* Training the Models with Synthetic Minority Oversampling (SMOTE) Technique

After observing the target column class counts, it was understood that due to the high availability of records in one target class, the outcome faced a severe class imbalance. This class imbalance can make the predictions favour the biased class that might overlook other important failure predictions. Therefore, even though the model results had comparatively optimal values that are discussed thoroughly in section IV, considering the imbalance of the classes the Synthetic Minority Oversampling Technique (SMOTE) was used to oversample when training the model to avoid predictions being biased on a single class. SMOTE mechanism that was introduced by researchers (Chawla et al., 2002) is very popular when it comes to addressing imbalanced data problems.

### *E.* Training Data using Deep Learning Models

The following listed deep learning model architectures were used to train the data in this research. All the models were trained for 50 epochs having a batch size of 256 with the Adam optimizer since the best results were produced with the use of these parameters. To select the best model architecture to gain proper results, various model combinations were tested rigorously. Therefore the model architectures were tested by including various combinations of neural layers, dropout, and batchnormalization layers. The batch normalization technique uses a transformation to keep the output mean and standard deviation close to 0 and 1, respectively thereby normalizing the input (Chollet et al., 2015). Dropout makes neurons to be dropped out from the model thereby preventing model overfitting (Chollet et al., 2015).

#### *1) Basic Deep Learning Model*

A simple basic neural network was first used to train the data model.

#### *2) Basic Deep Learning Model Trained with SMOTE*

Since there was class imbalance present in the original data, in this instance the SMOTE technique was used to balance the data and trained with the above same basic deep learning model. The rest of the trained models, therefore, used the SMOTE mechanism during the model training phases.

#### *3) RNN Model Trained with SMOTE*

An extension of a traditional feedforward neural network that can handle a variable-length sequence input is a recurrent neural network (RNN) (Chung et al, 2014). The RNN manages the variable-length sequence by utilizing a recurrent hidden state, the activation of which is dependent upon the previous activation at each subsequent iteration. For modelling sequence data, such as time series or natural language, RNNs are a class of effective neural networks (Chollet et al., 2015). An RNN model was therefore trained with SMOTE as the next step of the training stage.

#### *4) GRU Model Trained with SMOTE*

(Chung et al, 2014) proposed a Gated Recurrent Unit (GRU) to enable each recurrent unit to adaptively capture dependencies of various time scales. The GRU model has gating units that modulate the flow of information inside the unit similar to the LSTM unit, but without having separate memory cells. The GRU network uses a more straightforward structure than LSTMs and is simpler to train (Chung et al., 2014). Instead of using an input, it uses the update gate and reset gate. The next model was therefore comprised of a combination of the GRU model applied with SMOTE.

#### *5) LSTM Model Trained with SMOTE*

Hochreiter and Schmidhuber created the Long Short-Term Memory Networks (LSTM) to address the issue with conventional RNNs and machine learning algorithms (Chung et al, 2014). It is a specific variety of recurrent neural networks that can address the vanishing gradient issue that RNNs encounter. Compared to GRU, the LSTM has more gates and parameters, which increases its flexibility and expressiveness but also increases its computational complexity. This was well proved when the model blended with LSTM with SMOTE mechanism took much time to train compared to the rest of the models.

### *F.* Metrics Used for Model Evaluation

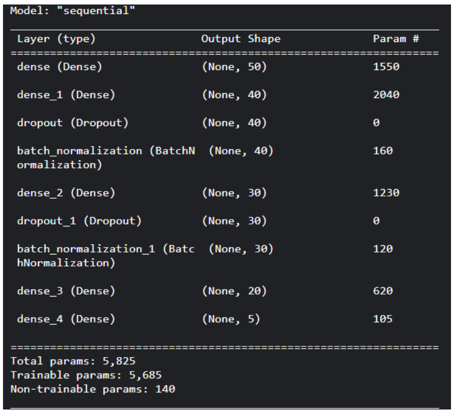
The metrics of macro average precision, macro average recall, and macro average f1-score also were used to compare the model performances. Due to the multi class behavior, the macro metrics were considered for evaluation purposes (Pedregosa et al., 2011).

To recognize a model with a good fit not only the metrics discussed above but the model loss values also should be considered (Brownlee, 2019). Therefore to assess the performances of the trained models, training and validation loss learning curves were also inspected using plots. Loss values can be derived using the learning curves that are calculated using the model's optimization metric (Brownlee, 2019). These loss curves are very useful when it comes to identifying overfitting in models.

## IV. Results

The resulting model summary of the basic deep learning model is shown in Figure 3. The rest of the models used the same architecture but with changes being made on the first layer as required for each case of RNN, GRU, and LSTM. Hence, the first layer of the deep learning model was replaced with RNN, GRU, and LSTM layers in the respective model while the rest of the model layers stood as the same. However, the model architecture of the best chosen model, which was the GRU-SMOTE that will be discussed thoroughly in section IV is displayed in Figure 4.

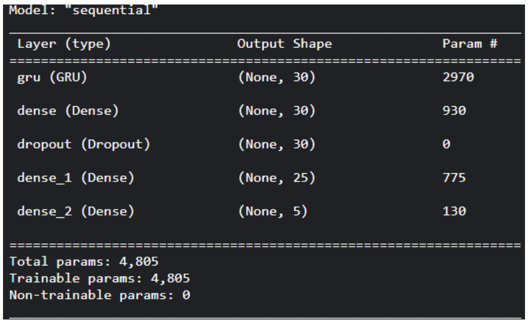
The accuracies recorded by each model architecture are depicted in Figure 5. To examine model overfitting, loss curves that were generated are shown in Figures 6-10, per each model. The metrics of macro average precision, macro average recall, and macro average f1-score were also examined for each and every model and recorded in Table 1 to select the best model.


Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense (Dense) | (None, 50) | 1550 |
| dense_1 (Dense) | (None, 40) | 2040 |
| dropout (Dropout) | (None, 40) | 0 |
| batch_normalization (BatchNormalization) | (None, 40) | 160 |
| dense_2 (Dense) | (None, 30) | 1230 |
| dropout_1 (Dropout) | (None, 30) | 0 |
| batch_normalization_1 (BatchNormalization) | (None, 30) | 120 |
| dense_3 (Dense) | (None, 20) | 620 |
| dense_4 (Dense) | (None, 5) | 105 |

Total params: 5,825
Trainable params: 5,685
Non-trainable params: 140

Figure 3. Model summary of the trained basic DL model


Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| gru (GRU) | (None, 30) | 2970 |
| dense (Dense) | (None, 30) | 930 |
| dropout (Dropout) | (None, 30) | 0 |
| dense_1 (Dense) | (None, 25) | 775 |
| dense_2 (Dense) | (None, 5) | 130 |

Total params: 4,805
Trainable params: 4,805
Non-trainable params: 0

Figure 4. The model architecture used for the GRU-SMOTE model

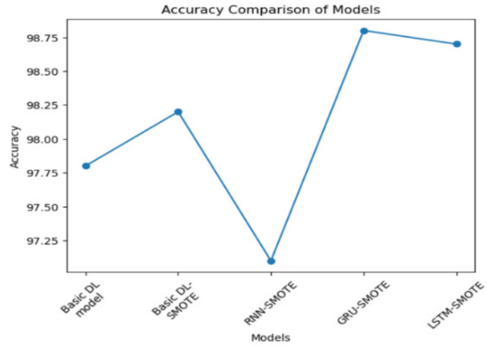



Figure 5. Accuracies recorded by each of the trained models.

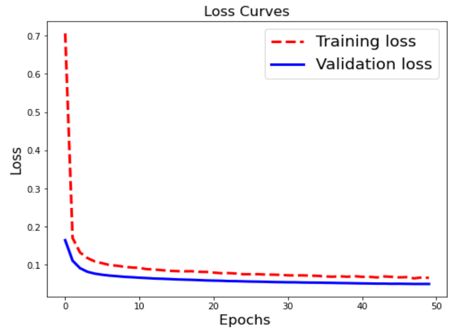



Figure 6. Loss graph for the model trained with a basic neural network

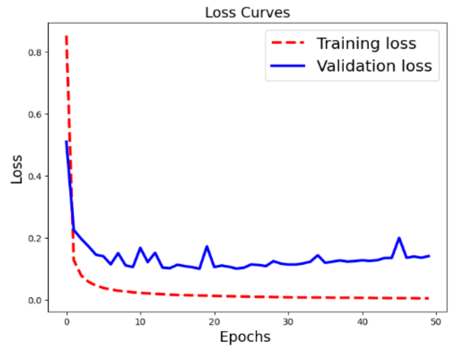



Figure 7. Loss graph for the model trained with SMOTE mechanism on the basic neural network

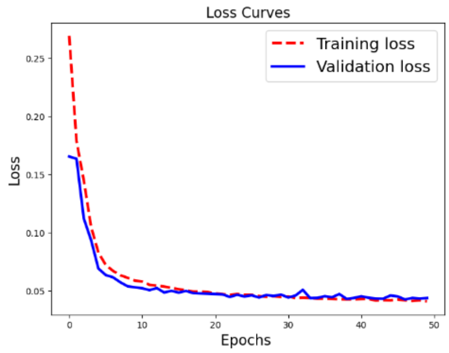



Figure 8. Loss graph for the model trained with SMOTE mechanism and GRU network

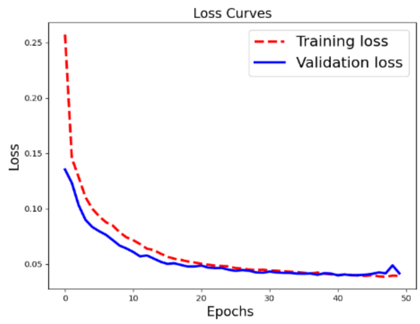



Figure 9. Loss graph for the model trained with SMOTE mechanism and RNN network

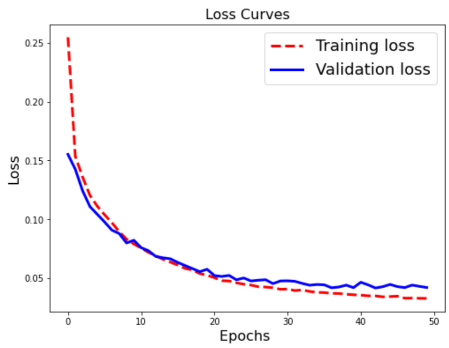



Figure 10. Loss graph for the model trained with SMOTE mechanism and LSTM network

Table 1. Evaluation metrics for trained models

| Model | Macro Average Precision | Macro Averag e Recall | Macro Average F1-score |
|---|---|---|---|
| Basic deep learning model | 0.93 | 0.69 | 0.79 |
| SMOTE based basic deep learning model | 0.70 | 0.73 | 0.72 |
| RNN-SMOTE model | 0.89 | 0.71 | 0.78 |
| GRU-SMOTE model | 0.91 | 0.70 | 0.78 |
| LSTM-SMOTE model | 0.92 | 0.70 | 0.79 |

## V. Discussion

According to Figure 5, it can be understood that all the trained models have been recorded with accuracies around the same range indicating all models have trained well. The models have high accuracies ranging from 97.1% to 98.77%. This indicates that the models are able to classify machinery failures with a high degree of accuracy. However, the model of GRU-SMOTE has achieved the highest accuracy of 98.7%.

When considering loss curves, the loss curve of the basic neural network trained with SMOTE given in Figure 7 indicates a clear case of overfitting due to the gap observed between the training and validation curves. It should be also noted that the LSTM-SMOTE model's loss curve illustrated in Figure 10 might also portray a slight amount of overfitting as during the ending phase of the training, the curves have started diverging. The loss curve of the RNN-SMOTE model in Figure 9 also has recorded a spike while ending the epochs training but has retained the original state at the end.

Hence, according to Figure 6, Figure 8, and Figure 9, the basic deep learning model, GRU-SMOTE model, and RNN-SMOTE model suggest a better control over overfitting because their respective loss graphs show a closer alignment between the training and validation losses. Although the loss curve of the basic deep learning model as given in Figure 6 has pointed out less overfitting, the model has no ability to handle the data imbalance issue present in the data which made the model be eliminated when choosing the finest model.

Therefore, only the models of RNN-SMOTE, and GRU-SMOTE, are the remaining contenders for the best model. Out of the remaining models of RNN-SMOTE and GRU-SMOTE, it can be seen that the GRU-SMOTE model has recorded slightly better scores for precision, recall, and f1-score than RNN-SMOTE model according to Table 1 while having the highest model accuracy according to Figure 5. Therefore the GRU-SMOTE model is the strong contender as the best model for this multi-class classification task as a result of these factors taken together. Table 2 summarizes all the five models with their capabilities highlighting the GRU-SMOTE model to be the finest.

Table 2. Characteristics of the models

| Model | Scores of, Accuracy Precision Recall F1-score | Handles Data Imbalance | Overfitting Present in the Model |
|---|---|---|---|
| The basic deep learning model | Optimal | No | No |
| SMOTE based basic deep learning model | Not optimal | Yes | Yes |
| RNN-SMOTE model | Optimal | Yes | No |
| GRU-SMOTE model | Optimal and has the best values | Yes | No |
| LSTM-SMOTE model | Optimal | Yes | Slight overfitting present |

To further investigate model overfitting, cross-validation was performed on splits of data using the GRU-SMOTE model which produced a mean accuracy of 98%. The successful implementation of the chosen GRU-SMOTE model opens up possibilities for proactive maintenance strategies, enabling timely interventions and reducing costly breakdowns.

## VI. CONCLUSION

In conclusion, this research work explores the fusion of predictive maintenance and deep learning models to address machinery maintenance challenges. By leveraging multi-class classification models, specifically Gated Recurrent Neural Networks, and techniques like SMOTE for data imbalance, accurate predictions of potential machinery failures were achieved. During the research not only standard metrics but also model loss curves were also inspected to choose the best model considering overfitting aspects. Using alternative techniques for handling data imbalance, such as cost-sensitive learning or different oversampling methods can also improve the accuracy and applicability of the predictive maintenance system considering future work. In conclusion, the application of multi-class classification models for predictive maintenance offers significant potential in improving maintenance practices and minimizing downtime. The chosen GRU based SMOTE model demonstrates its effectiveness in foreseeing equipment failures, allowing for prompt interventions. Furthermore, studies could concentrate on investigating cutting-edge architectures and incorporating extra data sources to improve the models' predictive abilities further.

## REFERENCES


Brownlee, J. (2019). A Gentle Introduction to Learning Curves for Diagnosing Machine Learning Model Performance. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research, 16, pp.321-357.

Chollet, F., & others. (2015). Keras. https://keras.io. https://keras.io

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv preprint arXiv:1412.3555.

Dey, R. and Salem, F.M., 2017, August. Gate-variants of Gated Recurrent Unit (GRU) Neural Networks. In 2017 IEEE 60th International Midwest Symposium on Circuits And Systems (MWSCAS) (pp. 1597-1600). IEEE.

Dheerasinghe, R. (2009). Garment Industry in Sri Lanka Challenges, Prospects and Strategies. Staff Studies, 33(1), 33–72.

Guduru, R. R., Shaik, S. H., Yaramala, S., TMS, N. P., & Domeika, A. (2018). A Dynamic Optimization Model for Multi Objective Maintenance of Sewing Machine. International Journal of Pure and Applied Mathematics, 118(20), 33–43.

Kanawaday, A. and Sane, A., 2017, November. Machine Learning for Predictive Maintenance of Industrial Machines Using Iot Sensor Data. In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 87-90). IEEE.

Patel, A. (2018). Problem Description. [online] GitHub. Available at:

https://github.com/ashishpatel26/Predictive_Maintenance_using_Machine-Learning_Microsoft_Casestudy [Accessed 4 Jul. 2022].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Serradilla, O., Zugasti, E., Rodriguez, J. and Zurutuza, U., 2022. Deep Learning Models for Predictive Maintenance: A Survey

Comparison, Challenges and Prospects. Applied Intelligence, 52(10), pp.10934-10964.

Susto, G.A., Schirru, A., Pampuri, S., McLoone, S. and Beghi, A., 2014. Machine learning for Predictive Maintenance: A Multiple Classifier Approach. IEEE Transactions on Industrial Informatics, 11(3), pp.812-820.

Wahid, A., Breslin, J.G. and Intizar, M.A., 2022. Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework. Applied Sciences, 12(9), p.4221.

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine Learning in Manufacturing: Advantages, Challenges, and Applications. Production & Manufacturing Research, 4(1), 23–4


## AUTHOR BIOGRAPHIES

UMMPK Nawarathne is an Academic Instructor at the Computing Centre, Faculty of Engineering, University of Peradeniya. Her main research interest lies on areas of Artificial Intelligence and Data Science.

Dr. Chamila Walgampaya is the Director of the Computing Center at the Faculty of Engineering, University of Peradeniya. He is also a Senior Lecturer at the Dept. of Engineering Mathematics. He earned his MSc and PhD degrees from the University of Louisville, USA. He has authored numerous journal and conference articles. His research interests are in Click fraud mining, Automatic web robots, Data fusion, Ensemble methods, and Machine learning.

Iroshan Aberathne is an Architect in Software Engineering and Data Science. He is reading for his Ph.D degree in Artificial Intelligence at Lincoln University, New Zealand after completing his M.Phil and B.Sc(Hons.) degrees at University of Peradeniya, Sri Lanka. He has 10+ years of experience as software architect, principal data scientist and senior lecturer. His industry experience and research interest mainly focuses on data science, machine learning, computational modelling, software engineering, and cloud computing.