IEEE BigData 2021 Cup: Soft Sensing Classification using LSTM Approach

Leelaprasad P
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
01fe19bcs278@kletech.ac.in

Siri Revant
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
01fe19bcs126@kletech.ac.in

Shreesh Bhat
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
01fe20bcs411@kletech.ac.in

Dhananjay Kalburgi
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
01fe19bcs131@kletech.ac.in

Shankar Gangisetty
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
shankar@kletech.ac.in

P.G Sunita Hiremath
School of Computer Science & Engineering
KLE Technological University
Hubli, Karnataka, India
pgshiremath@kletech.ac.in

***Abstract* - The major challenge in the wafer manufacturing industry is classification of wafers using imbalanced data generated from multiple sensors. The task we undertake is developing a deep learning technique for wafer classification. We proposed an LSTM approach that has multiple hidden LSTM layers where each layer contains multiple memory cells for deeper analysis. The performance of the LSTM approach was measured using the AUC metric with a learning rate of 0.005 and training for 1000 epochs. After performing experimental analysis our approach obtained a best AUC score of 0.737 (test set) compared to baseline models.**

1. Introduction:

The goal of this challenge is to find the best machine learning models for classifying soft sensing data. Using learning models to forecast measurement outcomes based on soft sensing data to save time for measurement and gain better insights into the predictive usefulness of these sensors for product quality.

Application of Soft Sensing in various Machine Learning Models.

1.1. Temperature Prediction in Rotary Kiln calcining process using soft sensor model on improved Elman NN with Dimension-Reduction

The Rotary kiln is an essential component of pellet sintering. The pellet making system produces green pellets with a diameter of 9-16 mm. The real time operating temperature in the rotary kiln calcining process has been projected using an enhanced Elman NN with changeable Data Preprocessing.

Data samples were collected and partitioned into two parts for Elman NN training, with 3000 process datasets of secondary variable and corresponding temperature in the Rotary kiln calcining process chosen for training and 1500 process datasets used as the testing dataset.

To Enhance predictive Accuracy a Data Preprocessing Technique based on the integration of ISOMAP and LLE is used to eliminate the noise and inconsistent data from input data.

The soft sensor model based on SVM is compared to the improved Elman NN with variable dataset preprocessing based on ISOMAP and LLE.

The Improved Elman NN with variable data preprocessing outperforms other soft sensor models in terms of learning performance. The Improved Elman NN's learning ability has been improved to some extent, and the processed variable data preprocessing method is effective in removing redundant data and noises.

1.2. Development of Novel Soft Sensor using LSTM and NMI feature Selection.

In this paper, an innovative soft sensor with time series features was employed to simulate complicated and evolving industrial processes.

The LSTM training is performed using real-world datasets, and NMI is then used to select variables related to a particular target variable. The proposed algorithm removes one irrelevant variable at a time until all variables have been removed.

Following that, the variable selection path appears, and the algorithm selects the segment with the lowest predicted error. The proposed soft sensor was put to use in two real-world industrial processes. This approach's effectiveness and superiority are demonstrated by simulation and comparison with other algorithms

.

The developed soft sensor offers an extra and reliable monitoring tool for crucial variables and can be used to design model predictive control systems in the future.

2. Data:

The dataset consists of five files:
X_train, X_val, X_test, label_train, label_val

The X_train, X_val, and X_test files consist of the features of the samples.
The label_train and label_val files consist of the output labels.

There are 817 features, and these are classified into three types as follows:
One hot encoded integer value of 6 categorical features ranging from 0 to 726 columns.
Float features are scaled numerical features ranging from727 to 815 columns. The 816 column serves as a padding indicator.
Using the X_train and label_train, the machine learning model needs to be built, and to check its performance, it has
to be evaluated across the validation set (i.e., by using X_val and label_val). Also, we need to generate a label_test file using X_test and build the model using the train set.

3. Methodology:

The machine learning algorithms used for building models are as follows:

1. Logistic Regression
2. Decision tree
3. LSTM

3.1. Logistic Regression:

This is a binary classification algorithm. Logistic regression's goal is to estimate the probabilities of events, including establishing a link between attributes and the probabilities of certain outcomes. The logistic function is a sigmoid function that takes any binary or real input t and outputs a value between zero and one. And depending on confidence, the coefficients of the model will update. This model was trained by using default parameters (i.e., with 100 epochs, 0 verbose, solver = 'lbfgs', etc.).

The model's performance was good, with an accuracy of 99% for the training set and validation set. But for the same output, the AUC score was 0.59 and 0.51 for the training and validation sets, respectively.

The main reason for the failure was that the model predicted all the one's as one but failed to predict many of the zero's as zero.

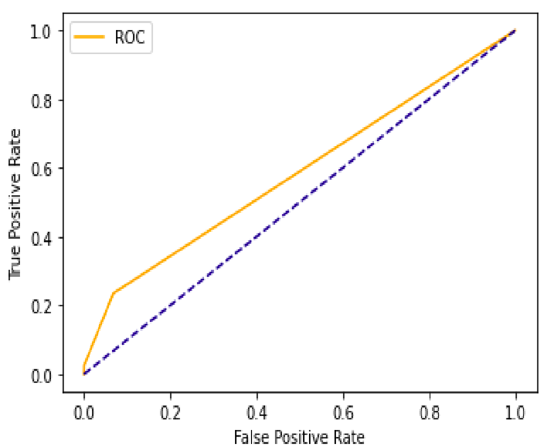

Fig 3.1.1 - Logistic Regression (AUC)

	tp	tn	fn	fp	tpr	fpr	min_dis	auc	score
0	0	1340	12	1	0	0.000745712	0.924675	0.511264	0.0123309
1	0	1395	5	0	0	0	1	0.462227	0.0115355
2	1	6717	9	5	0.1	0.000743826	0.250864	0.827556	0.0993103
3	0	1202	12	2	0	0.00166113	0.576792	0.615869	0.0133498
4	1	4132	89	39	0.0111111	0.00935028	0.978657	0.493455	0.037481
5	0	11106	68	0	0	0	1	0.469281	0.0934747
6	26	17458	393	361	0.0620525	0.0202593	0.820973	0.538506	0.175073
7	8	14	10	19	0.444444	0.575758	0.800761	0.466706	0.000424293
8	0	4052	3	4	0	0.000986193	0.501674	0.72929	0.0527682
9	0	1777	4	0	0	0	1	0.496924	0.0157763
10	1	5801	27	4	0.0357143	0.000689061	0.753168	0.591373	0.0614903

Fig 3.1.2 - Logistic Regression Evaluation Measures

3.2. Decision tree Classifier:

This is a classification algorithm. The purpose of employing a decision tree is to develop a training model that can be used to forecast the target variable's class or value by learning simple decision rules inferred from prior data. A decision tree does not require the scaling of data as well. The training period is shorter as compared to other commonly used models because it generates only one tree. This model was trained by using parameters like epochs = 100, verbose = 0, criterion = Gini, splitter = best, etc.

The performance of the model was good, with an accuracy of 89% for the train set. But for the same output, the AUC score was 0.57 and 0.51 for the train and validation sets, respectively.

The reason for the failure was the same. Since the given dataset is imbalanced, the prediction output skews towards a class label with a value of "Zero". Hence, we found two techniques to be the most feasible, namely, oversampling and undersampling. As the given dataset is large, choosing oversampling requires hardware with higher specifications (like RAM and GPU). Hence, we chose to go with under sampling.

The performance of the model using under sampled data was improved, with an accuracy of 92% for the train set. Also, for the same output, the AUC score was 0.63 and 0.54 for the train and validation sets, respectively. But no further improvement was found with any hyper parameter tuning.

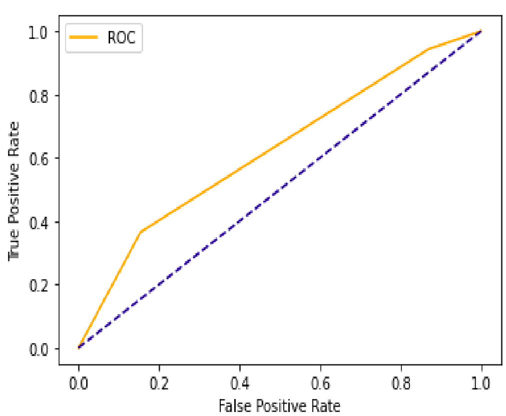

Fig 3.2.1 - Decision Tree Classifier (AUC)

	tp	tn	fn	fp	tpr	fpr	min_dis	auc	score
0	218	5892	46	13	0.825758	0.00220152	0.170449	0.912433	0.0170449
1	11	9892	16	26	0.407407	0.0026215	0.486374	0.744094	0.0224084
2	21	40599	93	88	0.184211	0.00216285	0.468312	0.742073	0.0916845
3	4	8822	97	46	0.039604	0.00518719	0.763238	0.53204	0.01445
4	10	30589	336	422	0.0289017	0.0136081	0.787921	0.57326	0.0544333
5	18	59710	544	24	0.0320285	0.000401781	0.768804	0.584261	0.106678
6	99	104124	1349	1497	0.0683702	0.0141733	0.799969	0.549629	0.178201
7	162	227	25	270	0.86631	0.54326	0.652847	0.622942	0.00129027
8	25	22216	47	10	0.347222	0.000449924	0.546587	0.723416	0.0488464
9	0	7844	24	0	0	0	0.500015	0.748095	0.0178238
10	96	34518	103	61	0.482412	0.00176408	0.455316	0.761757	0.080223

Fig 3.2.2 - Decision Tree Classifier Evaluation Measures

3.3. LSTM:

LSTM is a novel recurrent network architecture trained with an appropriate gradient-based learning algorithm. Here we chose to use stacked LSTM (Baseline). A single hidden LSTM layer precedes a normal feedforward output layer in the original LSTM model. The Stacked LSTM is a variant of this paradigm with many hidden LSTM layers, each with numerous memory cells.. Stacking LSTM hidden layers makes the model deeper, more accurately learning the description as a deep learning technique. This model was trained by using parameters like epochs=1000, verbose=0 , 3 layers(256,128,64), batch size=2048, learning rate=0.005 etc.).

The performance of the model was much improved, with an accuracy of 84% for the train set. Also, the AUC scores were 0.91 and 0.71 for the train and validation sets, respectively.

After the hyperparameter tuning for this algorithm, we found the best parameters for now. And this model was trained by using these parameters like epochs=1000, verbose=0, 4 layers (256,128,64,32), batch size=1024, learning rate=0.005 etc.).

The final performance of the model was much improved with an accuracy 74% for the train set. And, the AUC scores were 0.89 and 0.79 for train and validation sets respectively.

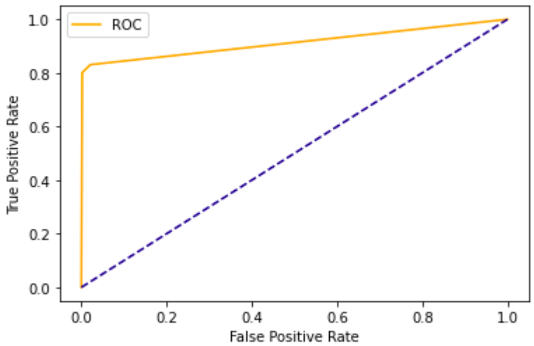

Fig 3.3.1 - LSTM (AUC)

	tp	tn	fn	fp	tpr	fpr	min_dis	auc	score
0	216	5532	56	488	0.794118	0.0810631	0.220276	0.916787	0.0161698
1	28	8739	5	1549	0.848485	0.150564	0.188257	0.903897	0.026151
2	109	40794	91	2195	0.545	0.0510596	0.337569	0.807092	0.0977112
3	106	8059	26	3055	0.80303	0.274879	0.313944	0.867915	0.0273605
4	393	23301	35	9493	0.918224	0.289474	0.250031	0.896684	0.0835051
5	414	59043	295	4964	0.583921	0.077554	0.38504	0.801378	0.145377
6	1456	69451	246	47881	0.855464	0.408081	0.378384	0.80917	0.269997
7	344	1267	99	481	0.776524	0.275172	0.307559	0.829452	0.00509427
8	58	19031	28	3389	0.674419	0.15116	0.327075	0.845683	0.0533524
9	39	6730	9	1144	0.8125	0.145288	0.20872	0.905591	0.0201101
10	172	29114	55	6760	0.757709	0.188437	0.297813	0.854929	0.0865163

Fig 3.3.2 – LSTM Evaluation Measures

4. Results:

Model	AUC	FPR	Accuracy
Logistic Regression	0.59	0.0022	0.99
Decision Tree Classifier	0.63	0.0004	0.92
LSTM	0.91	0.5432	0.84

From all the above model to model comparisons, the observation was that as the accuracy of the model decreased, the AUC score of the model was increasing.

In the Logistic Regression Model and the Decision Tree Classifier, even though accuracy was very high, recall was less than 30 percent. Hence, the AUC score observed was very low.

On the LSTM, the accuracy was less than the other two models, but the recall was in the range of 71-85 percent.

Hence, the AUC score was much improved.

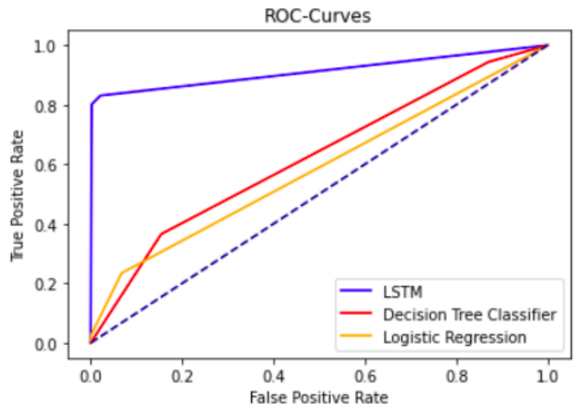

Fig 4.1 - Decision Tree Classifier (AUC)

In the above figure the ROC plots of all the three models were plotted.

LSTM being the highest with AUC of 0.91. Followed by Decision Tree Classifier having the next highest AUC of 0.63. And Logistic Regression being lowest with AUC of 0.59.

The Calculated False Positive Rate for the above models are 0.0004, 0.0022 and 0.5432 for logistic regression, Decision tree classifier and LSTM respectively.

5. Industrial Applicability:

We introduced a novel approach to classifying wafer quality during silicon wafer fabrication using LSTM Neural Networks in this paper. As many industries move toward automation, the wafer manufacturing business faces a hurdle in classifying wafer quality only based on characteristics such as pressure and temperature. The majority of approaches are limited to simulation-based trials, with only a handful contributing to industrial applications. Our LSTM approach can be utilised in this regard to boost industry productivity and scale up production speed while preserving or even improving classification precision. The Model can be used in the industry to analyse wafer quality in real time and can greatly minimise the time it takes to manually classify each wafer. This technology can be applied to industries that produce semiconductors, such as computer chips, storage devices, processors, electronic devices, and integrated circuits.

6. Conclusion:

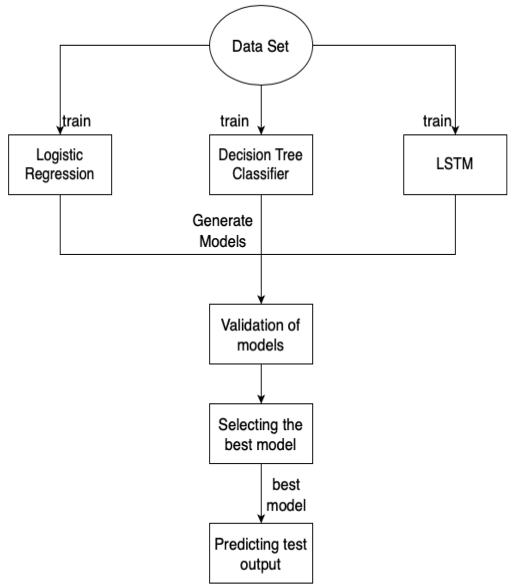

Fig 1. Process flow

The work mainly comprises of 3 steps:
Training - Training using all the three algorithms listed above.
Validation - Validation of data using all the three learning models built using training data.
Testing - Predicting test labels using the best learning model of all the three models.

7. References:

[1] Dongfeng Li, Zhirui Li, Kai Sun, "Development of a Novel Soft Sensor with Long Short-Term Memory Network and Normalized Mutual Information Feature Selection", *Mathematical Problems in Engineering*, vol. 2020, Article ID 7617010, 11 pages, 2020.

[2] Hai-bo Zhu, Yong Zhang, "Soft Sensor Model Based on Improved Elman Neural Network with Variable Data Preprocessing and its Application", *Journal of Sensors*, vol. 2018, Article ID 2981428, 10 pages, 2018

[3] Weiwu Yan, Huihe Shao, Xiaofan Wang,
Soft sensing modeling based on support vector machine and Bayesian model selection,
Computers & Chemical Engineering,
Volume 28, Issue 8, 2004, Pages 1489-1498, ISSN 0098-1354,

[4] X. Yuan, Z. Ge, B. Huang, Z. Song and Y. Wang, "Semisupervised JITL Framework for Nonlinear Industrial Soft Sensing Based on Locally Semisupervised Weighted PCR," in IEEE Transactions on Industrial Informatics, vol. 13, no. 2, pp. 532-541, April 2017, doi: 10.1109/TII.2016.2610839.

[5] Xuefeng Yan, Jie Wang, Qingchao Jiang,
Deep relevant representation learning for soft sensing, Information Sciences,Volume 514,2020,Pages 263-274,ISSN 0020-0255

[6] Zhiqiang Ge, Furong Gao, Zhihuan Song,
Mixture probabilistic PCR model for soft sensing of multimode processes,
Chemometrics and Intelligent Laboratory Systems, Volume 105, Issue 1,2011,Pages 91-105,ISSN 0169-7439

[7] Xiaofeng Yuan, Chen Ou, Yalin Wang, Chunhua Yang, Weihua Gui,
Deep quality-related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE,
Neurocomputing,Volume 396,2020,Pages 375-382,ISSN 0925-2312

[8] P. Kadlec and B. Gabrys, "Adaptive on-line prediction soft sensing without historical data," The 2010 International Joint Conference on Neural Networks (IJCNN), 2010, pp. 1-8, doi: 10.1109/IJCNN.2010.5596965.

[9] Mengüç, Y., Park, Y.L., Pei, H., Vogt, D., Aubin, P.M., Winchell, E., Fluke, L., Stirling, L., Wood, R.J. and Walsh, C.J., 2014. Wearable soft sensing suit for human gait measurement. *The International Journal of Robotics Research*, *33*(14), pp.1748-1764.

[10] Wang, G., Jia, Q.S., Zhou, M., Bi, J., Qiao, J. and Abusorrah, A., 2021. Artificial neural networks for water quality soft-sensing in wastewater treatment: A review. *Artificial Intelligence Review*, pp.1-23.