Learning from Sensor Data

SICK 3: Changlong Jiang, Yangjiang Dong, Gaomeizhu Qu, Yuxuan Su
{cljiang,dongyj,sylviaqu,suyuxuan}@bu.edu

1. Project Task

Our project is to investigate the relationship between different attributes and learn some potential rules from the dataset provided by SICK. Our goals include analysing relationship between package information trends and labels such as Legal For Trade (LFT) and Not Legal For Trade (NotLFT), belt speed impact on LFT and NotLFT, detecting outliers from gap, learning the indicative of "Pileup" situations from labels, and identifying the outliers in the data which caused by certain errors.
To sum up, We need to build a classification model whose output label is LFT and the input attributes are other conditions and information. We also need to detect outlier in gap and other data.

2. Related Work

Since we need to evaluate the information behind the trend and build a classifier for LFT label, we first implement some classic classification algorithms such as logistic regression, SVM, MLP and random forest via scikit-learn. Also, for adding time series information of packages (packages trend information), we consider our classification problem as a time series problem with multiple input variables which we found some experience [1]. In the meantime, we also research some other algorithms for time series such as Hidden Markov Model for time series prediction, which is introduced and explained in [2]. We also learn how to tune the LSTM hyperparameter in this [2].

3. Approach

3.1 Baseline Approach

Our baseline approach is using some classification algorithm for non-time-series input such as logistic regression, random forest, SVM and MLP. We use information of current package as input and 'LFT' label of current package as output. We use scikit-learn library to implement these algorithms.

3.2 Final Approach

For classification problem with multiple time-series attributes, we implement LSTM algorithm as our final model. We first transfer our time series data to structured supervised data. We take all information and conditions of precious 10 packages except 'Valid_Read' and 'Valid_Dim' condition of the current package. Our output is 'LFT' label of the current package. LSTM model structure is described below.

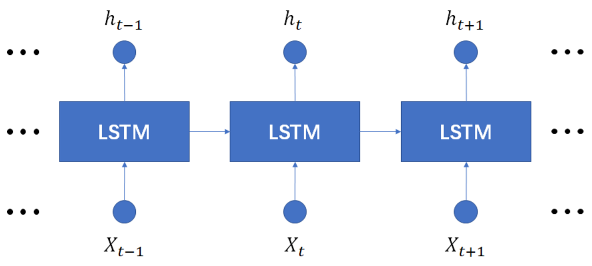

Figure.1 LSTM model

Since our problem is a binary classification problem, we use binary cross-entropy as our loss function which is described below:

$$L_{loss} = -(y log(p) + (1-y)\, log(1-p))$$

Here notice, *y* is binary indicator (0 or 1) if 'LFT' label is correct classification for observation, *p* is predicted probability observation is of class *'LFT'*.
Our input array is 207-columns which include all the information of previous 10 packages. Our output is the 1-D array describe the 'LFT' label of current package.
We use Keras library with tensorflow end to implement our LSTM.

4. Dataset and Metric

In our dataset, each package data has plenty of attributes, including package size, package weight, belt speed and package conditions. For pre-processing, we constructed different shape of dataframe from object csv file, then unified units, dropped useless attributes and got different shape of numpy array for multi-purpose. We got rid of much irrelevant information such as the case when oga $oga = -1000$ which means another condition named side by side.

To investigate the correlation between the "pile up" situation and package condition trends, we try to create one since there is no "pile up" condition label in the dataset,. if packages continuously have "Gap" label and if the package has side by side condition we would say they are piled up packages.

Finally, we implemented normalization on our data to the range of [-1:1]. Totally we have almost 1.5 million data, we use 50,000 data at the end of our dataset to test our algorithm. Therefore, the ratio of training data and test data is approximately 28/1.

The metrics that we used is F1 score, where the F1 score is defined as:

$$\frac{1}{f1score} = \frac{1}{precision} + \frac{1}{recall}$$

We use Classification metrics from scikit learn as baseline method.

5. Evaluation

We first implement our baseline approach to get more insights about package trend information behind the data. We find out the 'LFT' label is directly related to 'Valid_Read', 'Irreg' and 'Valid_Dim' label of current package. Here's result of our baseline model with 'Valid_Read', 'Valid_Dim' and 'Irreg'.

Evaluation	logistic regression	Random forest	SVM	MLP
F1 score	1.0	1.0	1.0	1.0

Table.1 Baseline Method f1 score with 'Valid_Read', 'Valid_Dim' and 'Irreg'.

Then we remove these three labels because our SICK partner states that these three labels might be useless to evaluating 'LFT' label. Here's result of our baseline model without these three labels.

Evaluation	logistic regression	Random forest	SVM	MLP
F1 score	0.80	0.82	0.80	0.81

Table.2 Baseline Method f1 score

For our time-series LSTM model. We start with using the information of previous 9 single packages and current package without these three labels. At this moment, we tune our hyperparameter epoch=10, neurons =10, batch size=32, the F1 score here is 0.9577 which is already higher than our baseline algorithm.

We then try several combinations of hyper-parameters to find the best combinations. The comparisons of hyper-parameters is shown in Table 2.

	epoch=10 neurons=10 batch size=32 previous packages=10	epoch=20 neurons=10 batch size=32 previous packages=10	epoch=20 neurons=20 batch size=64 previous packages=10	epoch=20 neurons=10 batch size=64 previous packages=10	epoch=20 neurons=10 batch size=64 previous packages=20
F1 score	0.9577	0.9673	0.9565	0.9812	0.9937
Precision	0.9221	0.9324	0.9292	0.9868	0.9879
Recall	0.9999	0.9999	0.9999	1	1
Loss	0.2926	0.1944	1.2268	0.0498	0.0355

Table.3 Hyper-parameter Evaluation

In the end, we choose epoch=20, neurons=10 , batch size=64 and previous packages=20, the loss and f1 curve is shown in figure 2, 3.

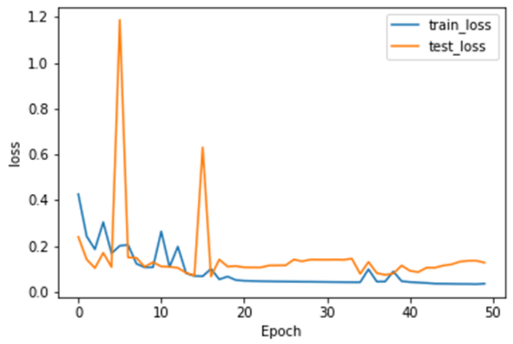

Figure.2. LSTM Model Loss

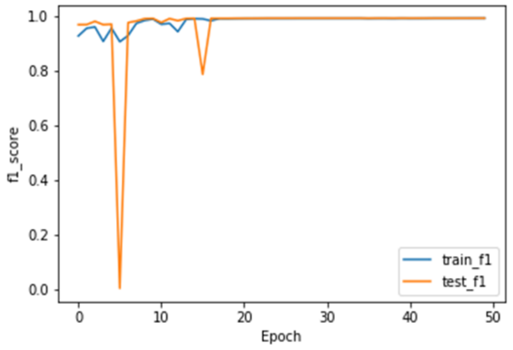

Figure.3. LSTM Model f1 score

In order to consider object time interval's effect to current model, we added another column called 'time-Interval' into the input data which is calculated by adjacent objects' 'timestamp'. Also we deleted the first object's data in each day to avoid the effect of abnormal 'time-Interval' data. the result is shown in figure 4, 5.

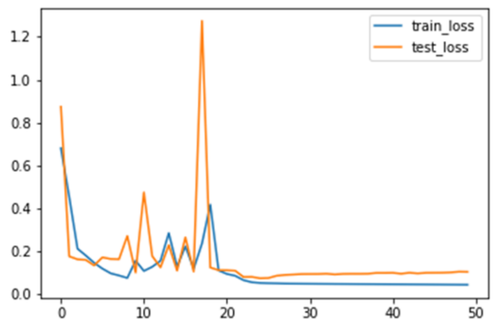

Figure.4. LSTM Model with Time-Interval Loss

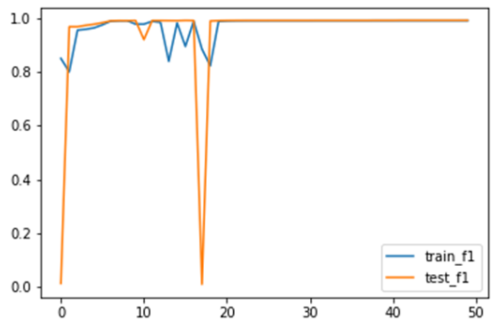

Figure.5. LSTM Model with Time-Interval f1 score

For the problem of analyzing of the belt speed, the belt speed are largely constant over time. According to our analysis, the average speed and variance over the days are listed below:

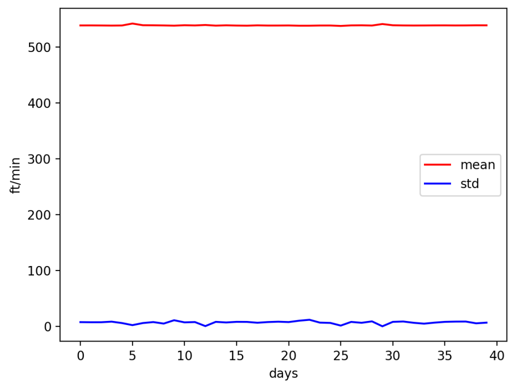

Figure.6 Mean and std speed over time

For getting the correlation of speed and LFT/NotLFT label, we utilize the pearson correlation coefficient, which is a measure of correlation for two mean-adjusted variables. Correlation coefficient of each day is calculated and the result is shown below.

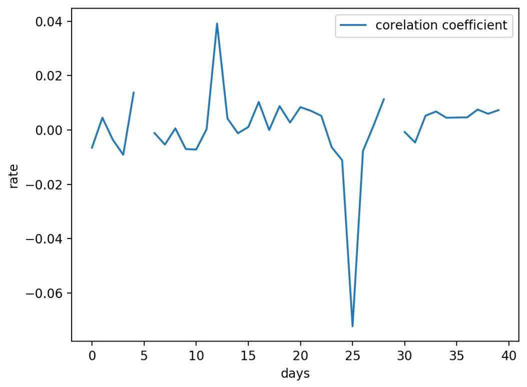

Figure.7 Correlation Coefficient of speed over LFT

According to this result, coefficients are all very close to 0, which indicates a low linear correlation between speed and LFT label. For a better understanding of the relationship between the two items, we also make use of decision tree models such as random forest to find the significance of speed to LFT.

6. Conclusion

We developed a LSTM model for ‘LFT’ classification with package trend information in this project. This model learn from time-series multiple variable we extracted from package trend and achieve around 0.99 F1 score in our final test, comparing with 0.82 in our baseline model. In the future, we will continue to explore how to deal with time-series data with imbalanced distribution of class label.

7. Roles

Task	**Lead**
Data pre-processing(350 lines)	Y. Dong, G. Qu, Y. Su,C. Jiang
Implement random forest and svm for LFT classification (250 lines)	Y. Dong,C. Jiang
Implement LSTM for LFT(600 lines)	C. Jiang,Y. Dong
Investigate correlation between “pile up” situation and package trends by using MLP and random forest, etc.(300 lines)	Y. Su
Outlier determination using gap information (“oga”) and investigate correlation between “Gap” situation and time intervals using SVM (300 lines)	G. Qu

8. Link to GitHub repo

Our Github link is :
https://github.com/dongyj1/Sick_Team_3

References

1).Multivariate Time Series Forecasting with LSTMs in Keras, Jason Brownlee. August 14,2017, https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

2).How to Tune LSTM Hyperparameters with Keras for Time Series Forecasting, Jason Brownlee on April 12,2017. https://machinelearningmastery.com/tune-lstm-hyperparameters-keras-time-series-forecasting/

3) Ch9, Speech and Language Processing. Daniel Jurafsky & James H. Martin. Draft of August 7, 2017.

Appendix:

1. Relevant tasks in project

1.1 Gap investigation

For outlier determination with Gap information and timestamp, we extract “oga” and timestamp data, where “oga” is labeled information for Gap. To get the boundary for Gap information specific, we implemented SVM and Robust linear model. First, we need to decide whether both “oga” and time interval information is considerable for analyze Gap. To get the relation between "oga" information and each adjacent objects' time interval, we use Robust linear model to fit the data.

The input of Robust model is x = (time interval, oga) . The output is co-efficiency of these two variables.

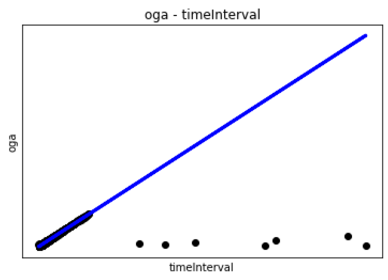

Figure.7 Robust Linear model

The result intends that it is not necessary to consider both time interval and "oga" for "Gap" investigation.

After plotting the "oga" and it's "Gap" condition, it is clearly separated into two parts along "oga" value. So we used SVM module to implement it. The input of SVM module is “oga” without negative data, which is split using cross_validation.train_test_split() function imported from Sklearn. The kernel of SVM is "linear" The output of this model is the classification. We calculate the threshold from the module by "threshold = intercept / coef”.

The result is as follows and the threshold of the model is 15.12967427.

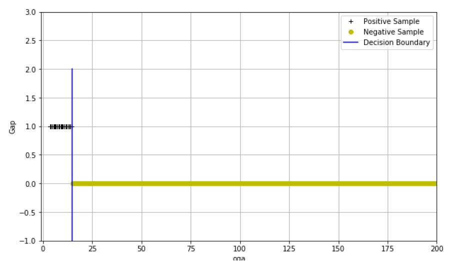

Figure.8 SVM model

For all the objects with "Gap" but not in the classification result, we call them detected outliers.

1.2 “Pile up” & “Log Jam”

Since there is no "Pile Up" & "Log Jam" label in the data, we need to identify "Pile Up" & "Log Jam" situation first. The gap between each package would be the main criteria to identify "Pile Up" & "Log Jam" situation. From gap information we know when gap is smaller than -1000, it is equal to "side by side" condition, so we would say all the packages whose gap is smaller than -1000 have "Pile Up" & "Log Jam" situation. Moreover, if "Gap" condition appears in package data, it means the gap distance between packages is smaller than 15 inches. So if "Gap" condition appears continuously, we would say those packages with "Gap" condition are in "Pile Up" & "Log Jam" situation.

the input of the model is all the “Irreg”, “MultiRead”, “TooBig” condition in the package data, the shape of the input is [1657178, 3], the output of the model is the predicted "pile up" condition, the shape of the input is [1657178, 1]. Since this is a classification problem, we use logistic regression and MLP and SVM with RBF kernel.

In logistic regression and MLP, we use Relu as activation function and the loss function is:

$$L(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} log h_\theta(x^{(i)}) + (1-y)^{(i)} log(1-h_\theta(x^{(i)}))]$$

where: $h_\theta(x) = log(1+exp(x))$ is the activation function,x is input and y is output.

Hyper-parameter: For logistic regression λ = 10, L2 regularization For MLP α = 0.0001, λ = 10, L2 regularization, hidden_layer_sizes = (20, 4)

In SVM with RBF kernel, the loss function is:

$$L(\theta) = C[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1-y)^{(i)} cost_0(\theta^T f^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

where: $f^i = exp(-\frac{||x - x^{(i)}||^2}{2\sigma^2})$ for gauss kernel,

Hyper-parameter: C = 2, $\gamma = 100$ where $\gamma = 2\sigma^{-2}$

After using the same training data and test data with LSTM, the f1 score of the “pile up” is shown below:

Methods	Logistic regression	MLP	SVM
F1 score	0.497984735186	0.497984735186	0.497984735186

Table.4. f1 score for “pile up” classification

Since the f1 score is nearly 0.5, therefore, the relationship between “pile up” and “Irreg”, “MultiRead”, “TooBig” is random and irrelevant.