Learning Data Quality Analytics for Financial Services

Completed Research Paper

Ka Yee Wong **Raymond K. Wong**

Haojie Huang

Abstract

Financial institutions put tremendous efforts on the data analytics work associated with the risk data in recent years. Their analytical reports are yet to be accepted by regulators in financial services industry till early 2019. In particular, the enhancement needs to meet the regulatory requirement the APRA CPG 235. To improve data quality, we assist in the data quality analytics by developing a machine learning model to identify current issues and predict future issues. This helps to remediate data as early as possible for the mitigation of risk of re-occurrence. The analytical dimensions are customer related risks (market, credit, operational & liquidity risks) and business segments (private, wholesale & retail banks). The model is implemented with multiple Long Short-Term Memory ("LSTM") Recurrent Neural Network ("RNNs") to find the best one for the quality & prediction analytics. They are evaluated by divergent algorithms and cross-validation techniques.

Keywords: Long Short-Term Memory, Recurrent Neural Network, CPG 235

Introduction

In Feb 2019, the government demanded for the restoration of trust in financial system after several mis-conduct of the banks (Frydenberg 2019). This is attributable to recent scandals: In April 2018, the Prudential Regulator refused a bank's corporate risk data due to data inaccuracy and incompleteness (Frost 2018). In this month, the Royal Commission challenged banks for financial mis-conduct arising from the poor quality of risk data (Yeates 2018).

The reality is that banks have the obligations to comply with the APRA CPG 235 released in 2013 (APRA 2013). CPG 235 sets out a technique for managing data risk – data quality dimensions. In recent years, many banks are striving to build an analytics hub to implement machine learning for high quality analytics (Crozier 2017).

Due to the poor quality of risk data, we develop a machine learning model for the risk data analytics – existing quality & future quality prediction. It enables the management to understand the room for

improvement and spot poor data in a forward-looking manner. Accordingly, they can remediate data earlier for the mitigation of risk of reoccurrence.

The model outputs analytical reports of data quality by customer related risks including market risk ("MR"), credit risk ("CR"), operational risk ("OR") & liquidity risk ("LR"), and business segments such as private bank ("PvB"), wholesale bank ("WB") & retail bank ("RB"). It is developed with a data quality scoring approach in alignment with the CPG 235 and implemented by deep learning networks – LSTM RNNs.

The selection for these networks is to meet the regulatory requirement – scalability of analytical reports across years and the consideration of previous & future cases for the reduction of likelihood of a future occurrence. The reports for different risk data over years ought to be made available upon receipt of an ad-hoc request from regulator. This requires a massive network. The analytics capability is expected to extend to prediction.

LSTM RNNs can be used to classify data over years efficiently and develop a forward-looking ability to forecast issues. They model long-term temporal dependencies automatically (Zhu et al. 2016) and exploit information from the past & future (Zhou et al. 2016) to build a giant network. They can remember memory across long sequences to obtain control over when internal state is cleared and forecast issues to reduce the risk of similar issues.

Machine Learning Approach

According to the CPG 235, the data quality ("DQ") is to be assessed by the dimensions: (a) accuracy; (b) completeness; (c) consistency; (d) timeliness; (e) availability; and (f) fitness for use (APRA 2013). These are broken down into the quality issues as defined in Figure 1. In total, 10 quality issues (Xu 2002; Singh et al. 2010) are mapped to the dimensions. The fewer the issues, the higher the quality.

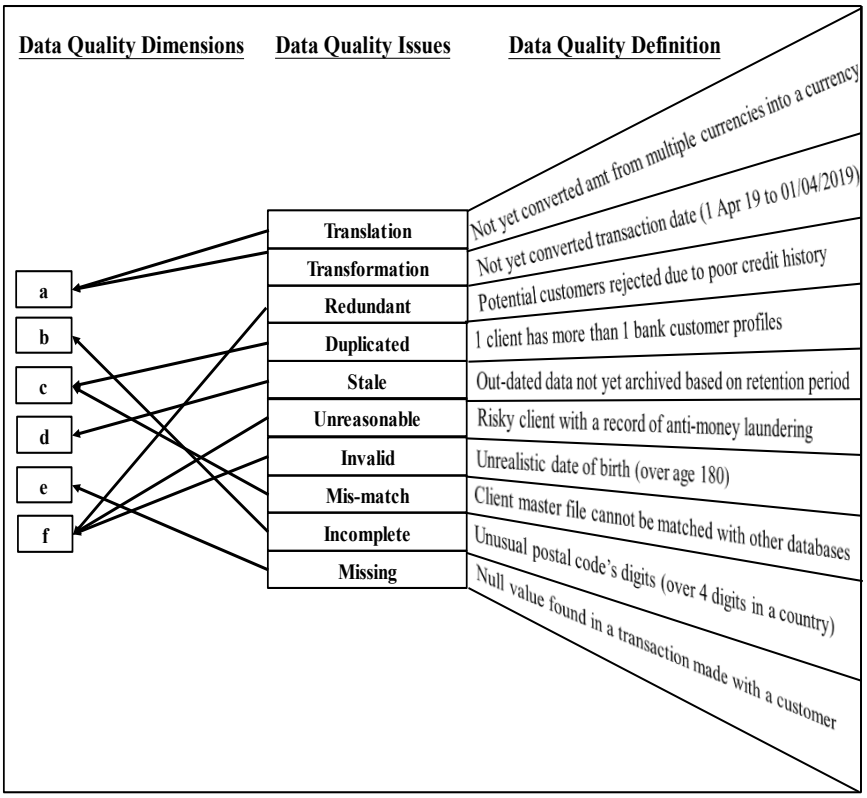


Figure 1. Data Quality Issues Mapped to the CPG 235 Data Quality Dimensions

The Dataset

Real-world data quality issues were regularly announced by risk experts (e.g. MR (KPMG 2018; CFI 2018), CR (Moody's Analytics 2018), LR (IOSCO 2018) and OR (Migueis 2018; Groenendijk et al.

2018). In this paper, we: a) analyze the structure of these and summarize key characteristics; b) capture the commonalities of issues to replicate a similar dataset; and c) magnify common issues in the dataset.

We synthesize 1 million banking customer records that capture all possible non-compliance scenarios according to the CPG 235. This dataset is data input in the model. It has 132 data features (called "data elements") belonging to 4 risk databases. Each database contains 33 features in which 8 are static and 25 are dynamic. Some features are extracted to Table 1. They include corporate & individual data, and the values are discrete instead of continuous.

Table 1. Data Features (Examples)

MR	CR	OR	LR
Asset Maturity (1945 days, tbc, na)	Loan ID (385623, 0, tbc, na)	Loss Income Ratio (1.15%, 92.04%, 54.6%)	Liquidity Rate (10.39%, 65.29%)
NPV (425543, 0, tbc, na)	Weighted Avg PD (6.31%, 19.48%)	Residual Legal Liability ($1385, 12, 0)	Instrument (TBC, Forward, Equity)

Data features are embedded with quality issues such as MR's NPV (0, tbc and na), CR's loan ID (0, tbc or na), OR's legal liability (0) and LR's instrument (TBC).

Data Labeling, Scoring and Pre-processing

Prior to inputting data into the networks, we labeled them as depicted in Figure 2.

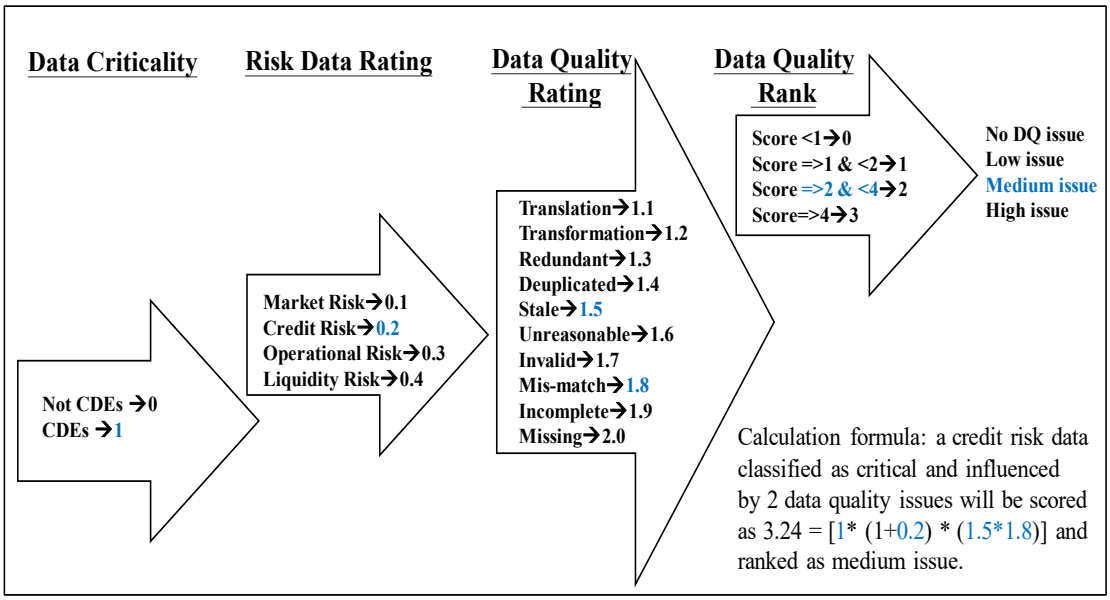


Figure 2. Data Labeling

We: a) label data as 1 or 0 to indicate if it is critical or not; b) assign a risk data rating (0.1 to 0.4) and a data quality rating (1.1 to 2) based on the types of risk data and quality issues respectively; c) then classify data quality scores (<1, =>1 & <2, =>2 & <4 and =>4) into four ranks (no/ low/ medium/ high data quality issue) and compare the ranking, actual output, with the prediction made in the experiment. The ratings are justified:

a) Data criticality: Data needs to be classified based on business criticality & sensitivity (APRA 2013). Referring to a research defining factors impacting data quality (Xu et al. 2005), we make similar assumption: $DE_{Criticality} = 0$ if data element ("DE") is not used in data aggregation and $DE_{Criticality} = 1$ if DE is used in the aggregation influencing data quality.

b) Risk data rating: different risk types are inherited with different levels of risk. MR is approximated at 10% ($E(R_i)$) under a CAPM (Hwang et al. 1999), CR is assumed to be 20% (CVaR) under a

confidence level of 99.5% to 99.99% for the finance sector using Monte-Carlo simulation (Dan et al. 2010), OR is set to 30% due to VaR between 27.84% and 37.71% (Allen et al. 2007; Shevchenko et al. 2006) and LR is defined as 40% (R^2) under a liquidity measure of Depth (Chordia et al. 2000; Wong et al. 2009);

c) Data quality rating: min. or max. operation (from 0 to 1) can be applied to aggregate multiple quality issues (Pipino et al. 2002). We define data quality ratings (1.1 to 2.0) by normalizing 10 issues. These are subjective since there is no empirical research available; and

d) Data quality scores: we classify scores after referencing to a paper ranking quality to allow the management to understand which ones are crucial to data quality (Xu et al. 2005).

The score is a multiplication of the rating for data quality issues. For overall quality, we compute: Data Criticality Factors*(1+Risk Data Rating)*Data Quality Rating.

This formula is derived based on: a) Multiplication of all factors: the more the quality issues, the more complicated the issues. The complexity is intensified by integrating divergent issues. This is called integrated quality rating (Kumar et al. 2005); and b) The risk data rating is a summation of 1: risk data has inherent risk which is additional to the data element.

We pre-process data by identifying abnormal data deviating from the standard (such as missing values) and normalizing data to a range of value, 0 to 1, by a min-max scaler.

The Model

In the model, 4 LSTM RNNs (Abdel-Nasser et al. 2017) are deployed for analyzing time series regression prediction. They help to analyze if the quality issues exceed data risk thresholds or not.

These 4 networks constitute the model as illustrated in Figure 3 below.

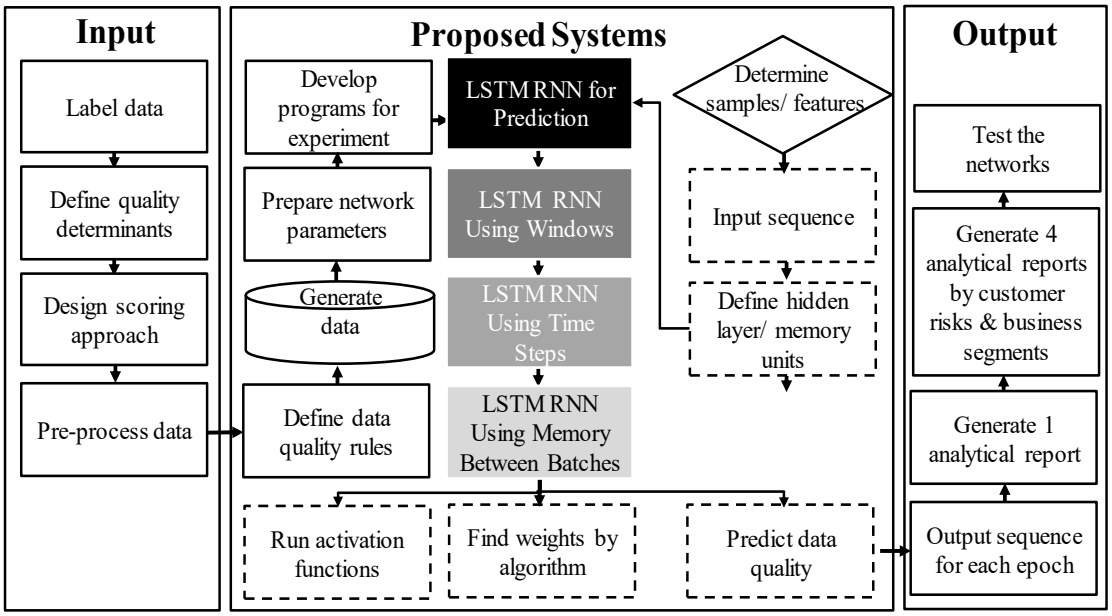


Figure 3. Model

The model preparation is development of programs for training the networks, generation of a dataset, implantation of quality issues into the data and the measurement of data quality score before data pre-processing. Four networks are trained to generate analytical reports for the existing quality issues and for the prediction of future issues by customer risks (in terms of MR, CR, OR & LR) and business segments (such as PvB, WB & RB).

The model frames divergent LSTM RNNs to predict data quality, as described in Table 2.

Table 2. Networks & Relevant Methodologies

Networks	Methodologies
LSTM RNN	Input data (X) is in the form of: samples, time steps, features. There is one sample & feature. Given the data quality issues for each data element now (t), we predict the problem for next time (t+1). For this time series data, we prioritize the sequences of values and define look_back – number of previous time steps as input variables to predict next result. In case the number is 1, next result will be t+1. Also, we define a layer with 1 input, a hidden layer with 4 LSTM blocks and 1 output layer. The activation function is sigmoid and the number of epochs is 10 while the size of batch is 1. After fitting data into the network, we make prediction based on training & testing data. Then we test the network for unforeseen data by cross-validation techniques.
LSTM RNN Using Windows	The data quality is predicted at next time (t+1) by utilizing current time (t) and two recent timesteps (t-1 and t-2) as inputs. The number of previous timesteps is a window and the size of it is tuned for each problem. By looking back, the error may increase and so the window size and network architecture will be tuned.
LSTM RNN Using Time Steps	Previous time steps are taken as inputs to predict output at next step instead of treating the past observations as separate input features. As such, different numbers of timestep are used – from a point of failure or a point of surge. This enables to know as to whether the problem is framed accurately or not.
LSTM RNN Using Memory Between Batches	Utilizing memory to make prediction can remember long sequences. When fitting data into network, the state will be reset after each batch. This allows to manage as to when the internal state of LSTM network is cleared. As a result, a stateful layer is formed. At the end, the state for complete sequence is developed. In training the network, no data is reshuffled, and the network state is reset after each epoch. Once the network is built, the stateful parameter is set to true. In setting the batch input shape, we hard-code the number of samples in a batch, the number of timesteps in a sample and the number of features in each time step. Thus, we forecast the issue to see if they exceed threshold.

We train multiple LSTM RNNs to find the most favorable one. They model varying length sequences and capture long range dependencies in the analytics of current quality & future prediction.

System Architecture

The topology for LSTM RNNs (Ergen et al. 2017) is outlined in Figure 4.

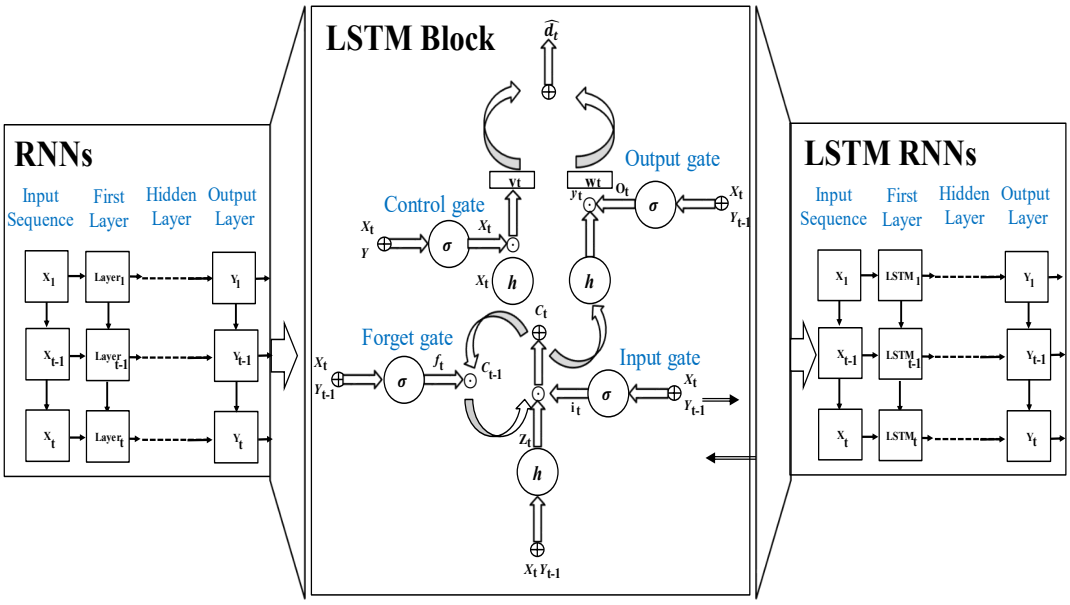


Figure 4. System Architecture

LSTM RNNs have memory blocks connected via layers (input, hidden & output layers). The blocks for recent sequences contain a block state & output. At input layer, the blocks start with input sequences ($x_1, x_{t-1}, \ldots x_t$) and their gates use sigmoid function (σ) to control if they modify the cell state. Gates are forget, input, output & control gates. Each has its weight (w_t) to learn. At output layer, a value of 1 or 0 is output to predict quality ($y_1, y_{t-1}, \ldots y_t$) indicating if the quality exceeds data risk thresholds.

Network and Relevant Algorithm

RNN generates vector sequence $\widehat{d_t}$ of hidden state by computing the equation (Ergen et al. 2017; Fan et al. 2014).

$$h_t = K\left(W^{(h)}x_t + R^{(h)}h_{t-1}\right)$$

$$y_t = u\left(R^{(y)}h_t\right)$$

where $h_t \in \mathbb{R}^m$ is state vector, $x_t \in \mathbb{R}^p$ is input, y_t is output, t is time and h_{t-1} is previous hidden state. The function $K(\cdot)$ and $u(\cdot)$ apply to vectors pointwise and set to $\tanh(\cdot)$ while the coefficient matrices are:

$$W^{(h)} \in \mathbb{R}^{m \times p}, R^{(h)} \in \mathbb{R}^{m \times m} \text{ and } R^{(y)} \in \mathbb{R}^{m \times m}$$

In LSTM, the network learns from the past and previous prediction of a specific timestep to make prediction. Input time starts from t-1, t to t+1 and input in network is a sequence of samples ($x_1, x_2, \ldots x_t$). It is passed to the 1st layer at a given time t ($t = 1, 2, \ldots T$). Given 1 hidden layer, the activation of units for memory blocks in layers is:

$$z_t = h(W^{(z)}x_t + R^{(z)}y_{t-1} + b^{(z)})$$

$$i_t = \sigma(W^{(i)}x_t + R^{(i)}y_{t-1} + b^{(i)})$$

$$f_t = \sigma(W^{(f)}x_t + R^{(f)}y_{t-1} + b^{(f)})$$

$$c_t = \Lambda_t^{(i)} + \Lambda_t^{(f)}c_{t-1}$$

$$o_t = \sigma(W^{(o)}x_t + R^{(o)}y_{t-1} + b^{(o)})$$

$$y_t = \Lambda_t^{(o)}h(c_t)$$

where $\Lambda_t^{(f)}$ is $\text{diag}(f_t)$, $\Lambda_t^{(i)}$ is $\text{diag}(i_t)$ and $\Lambda_t^{(o)}$ is $\text{diag}(o_t), c_t \in \mathbb{R}^m$ is state vector (generated by calculating the weighted sum using previous cell state & current information generated by the cell), $x_t\mathbb{R}^p$ is input vector, $y_t \in \mathbb{R}^m$ is output vector. The gates are input gate (i_t), forget gate (f_t) and input gate (o_t). This time, the function $g(\cdot)$ and $h(\cdot)$ apply to vectors wisepoint and set to $\tanh(\cdot)$ but sigmoid function $\sigma(\cdot)$ applies wisepoint to vector elements. Consequently, the co-efficient matrices & weight vectors:

$$W^{(z)} \in \mathbb{R}^{m \times p}, R^{(z)} \in \mathbb{R}^{m \times m} \text{ and } b^{(z)} \in \mathbb{R}^m$$

$$W^{(i)} \in \mathbb{R}^{m \times p}, R^{(i)} \in \mathbb{R}^{m \times m} \text{ and } b^{(i)} \in \mathbb{R}^m$$

$$W^{(f)} \in \mathbb{R}^{m \times p}, R^{(f)} \in \mathbb{R}^{m \times m} \text{ and } b^{(f)} \in \mathbb{R}^m$$

$$W^{(o)} \in \mathbb{R}^{m \times p}, R^{(o)} \in \mathbb{R}^{m \times m} \text{ and } b^{(o)} \in \mathbb{R}^m$$

Assuming output is y_t & w_t is final regression coefficient, the final estimate is:

$$\widehat{d_t} = w_t^T y_t$$

Input gate & forget gate govern the information flow into & out of the cell c_t while output gate controls how much information from the cell is passed to the output. Using current input, the state of previous step generated h_{t-1} and current state of the cell c_{t-1} decide if data inputs are taken, memory stored before is forgotten and the state of output is generated (Zhu et al. 2016; Zhou et al. 2016).

The main algorithm of the networks is ADAM. It is derived from adaptive moment estimation and computes dual adaptive learning rates for multiple parameters from the estimates of the first and second moments of the gradients. This is the reason why we choose this algorithm in comparison with other algorithms. It is computed as:

$$x_t = x \cdot \frac{\sqrt{1-\beta_2^t}}{(1-\beta_1^t)} \text{ and so } x_t = \theta_t \leftarrow \theta_{t-1} - x_t \cdot m_t / (\sqrt{v_t} + \hat{\epsilon})$$

where β is delay rate, t is time step, m_t is moving average of gradient, v_t is squared gradient, θ is parameter, Assuming $f(\theta)$ is an objective function, the stochastic scalar function is differentiable with regards to the parameter. To minimize the expected value of this, E[$f(\theta)$], we define the realization of stochastic function at timesteps 1,…, t and the gradient (vector of partial derivatives of ft) at timestep. In this case, the algorithm updates exponential moving averages of gradient and squared gradient whenever the hyper-parameters β1, β2 ∈ [0,1] control the exponential decay rates of these moving averages.

Experiment

Python v3.5 is used with Keras library & tensorflow to train the networks on a system with i.7-7500U CPU@2.9GHz, OS of 64-bit and Win 10 Pro. The data used can be accessed from http://ndb.cse.unsw.edu.au/regtech/datasets/201904. 70% of the data is fitted to the networks and 30% is used to evaluate the networks. Current quality issues are compiled before predicting the potential issues.

Results

To predict the data quality of integrated dataset, we train 4 networks with the algorithm of ADAM to output the prediction accuracy ("Acc") & error ("Loss"), as displayed in Table 3.

The accuracy for all RNNs is similar (at a level of 69%) in the 10th epochs but the level is consistently high only for the LSTM RNN using memory between batches. With regards to the loss, this LSTM RNN using memory is as good as the LSTM RNN using time steps. The loss for the former is minimized at the 1st and end of the epoch whereas that for the latter reaches a minimal level in last 3 epochs in comparison with others. Both seem to be comparable for the prediction of quality.

Table 3. Accuracy & Loss for 4 LSTM RNNs

LSTM	RNN		RNN using Win		RNN w Time Steps		RNN using Memory	
Epoch	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss
1	0.6913	0.6198	0.6877	0.6366	0.6918	0.6191	0.6921	0.6188
2	0.6921	0.6183	0.6921	0.6183	0.6921	0.6183	0.6921	0.6184
3	0.6921	0.6181	0.6921	0.6183	0.6921	0.6182	0.6921	0.6183
4	0.6921	0.6179	0.6921	0.6183	0.6921	0.6181	0.6921	0.6182
5	0.6921	0.6179	0.6921	0.6183	0.6921	0.6181	0.6921	0.6181
6	0.6921	0.6179	0.6921	0.6183	0.6921	0.6179	0.6921	0.6181
7	0.6921	0.6179	0.6921	0.6183	0.6921	0.6179	0.6921	0.6180
8…10	0.6921	0.6179	0.6921	0.6183	0.6921	0.6179	0.6921	0.6179

To confirm the prediction error, we estimate mean squared error ("MSE"), as exhibited in Figure 5. The lowest MSE is achieved by three RNNs including LSTM RNN, LSTM RNN using time steps and LSTM RNN using memory between batches. These three RNNs minimize the loss to 0.2133 over 10 epochs.

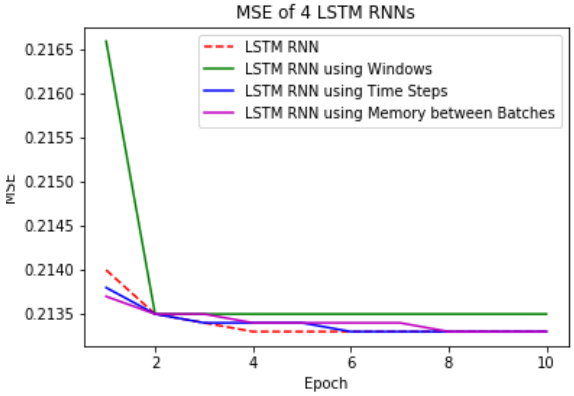
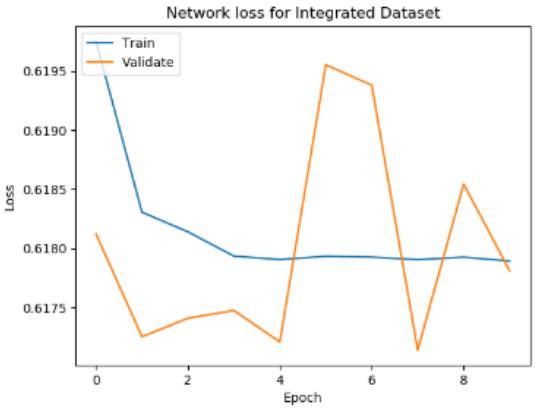


Figure 5. MSE for 4 LSTM RNNs

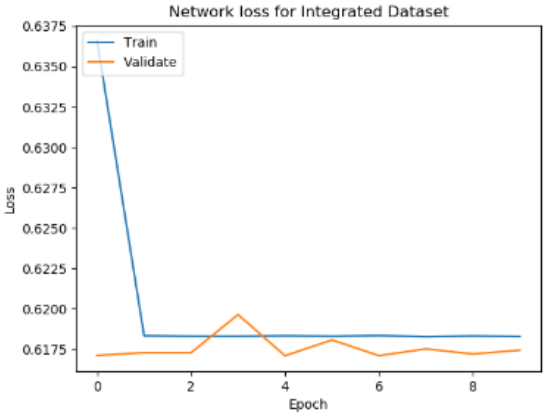
Evaluation

After the experiment, we test the effectiveness of RNNs and compare the results with actual output. The validated accuracy is equal for all RNNs (69.25%) & validated losses are made in Figure 6. The loss is the lowest in LSTM RNN using Windows (0.6174). Similarly, the validated MSE is minimized in this RNN (0.2131) out of all, as visualized in Figure 7. In view of this, LSTM RNN using Windows is superior to other RNNs. The MSE for others: 0.2132 (LSTM RNN), 0.2133 (LSTM RNN using time steps) and 0.2134 (LSTM RNN using memory between batches).

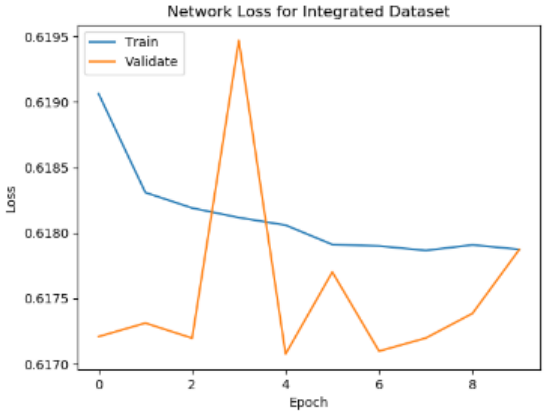
LSTM RNN



LSTM w/ Win



LSTM w/ Time Steps



LSTM w/ Memory

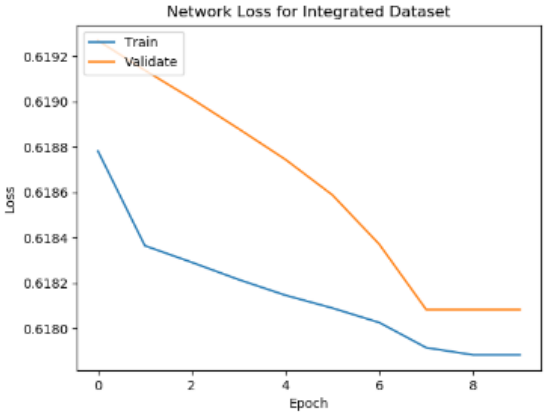


Figure 6. Validated Loss for 4 LSTM RNNs

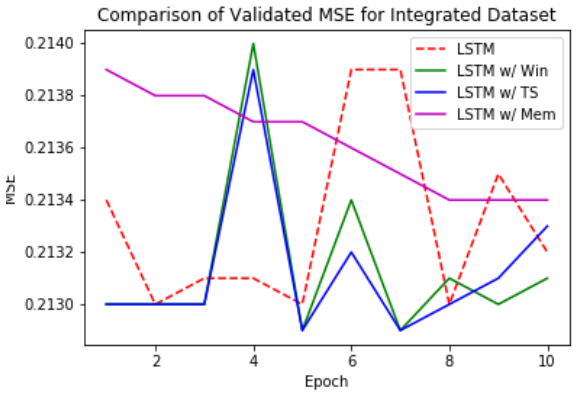


Figure 7. Validated MSE for 4 LSTM RNNs

Case Studies

For more analytics of the prediction, we study three cases below.

Case 1 – We select LSTM RNN using memory and LSTM with time steps for a further study as a result of the similar excellent performance. To maximize the accuracy and minimize the loss, we analyze their prediction by three more algorithms, as made in Tables 4 and 5. LSTM RNN using memory with ADAGRAD achieves the highest accuracy (69.21%) and the lowest loss (0.6174). As such, we prefer to use this LSTM RNN.

Table 4. Accuracy & Loss of RNNs with Time Steps under 4 Algorithms

RNN	ADAM		RMSPROP		ADADELTA		ADAGRAD	
Epoch	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss
1	0.6918	0.6191	0.6919	0.6267	0.6919	0.6218	0.6918	0.6183
2	0.6921	0.6183	0.6921	0.6219	0.6921	0.6222	0.6921	0.6177
3	0.6921	0.6182	0.6921	0.6217	0.6921	0.6228	0.6921	0.6176
4	0.6921	0.6181	0.6921	0.6216	0.6921	0.6221	0.6921	0.6176
5	0.6921	0.6181	0.6921	0.6216	0.6921	0.6221	0.6921	0.6175
6	0.6921	0.6179	0.6921	0.6216	0.6921	0.6221	0.6921	0.6175
7	0.6921	0.6179	0.6921	0.6217	0.6921	0.6221	0.6921	0.6175
8	0.6921	0.6179	0.6921	0.6216	0.6921	0.6213	0.6921	0.6175
9	0.6921	0.6179	0.6921	0.6216	0.6921	0.6214	0.6921	0.6175
10	0.6921	0.6179	0.6921	0.6216	0.6921	0.6215	0.6921	0.6175

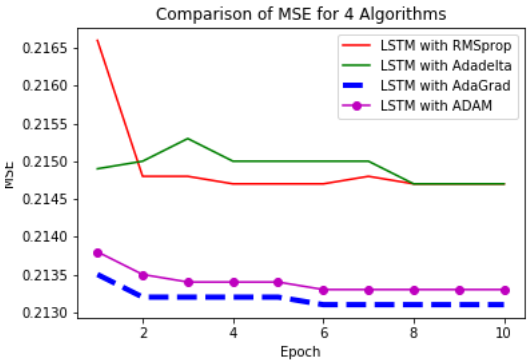
Table 5. Accuracy & Loss of RNNs using Memory under 4 Algorithms

RNN	ADAM		RMSPROP		ADADELTA		ADAGRAD	
Epoch	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss
1	0.6921	0.6188	0.6921	2.8424	0.6921	0.6213	0.6921	0.6178
2	0.6921	0.6184	0.6921	4.9079	0.6921	3.9912	0.6921	0.6175
3	0.6921	0.6183	0.6921	4.9079	0.6921	4.9079	0.6921	0.6175
4	0.6921	0.6182	0.6921	4.9079	0.6921	4.9079	0.6921	0.6175
5…6	0.6921	0.6181	0.6921	4.9079	0.6921	4.9079	0.6921	0.6174
7	0.6921	0.6180	0.6921	4.9079	0.6921	4.9079	0.6921	0.6174
8…10	0.6921	0.6179	0.6921	4.9079	0.6921	4.9079	0.6921	0.6174

We examine the MSE of these two LSTM RNNs, as outlined in Figure 8.

The loss is minimized in ADAGRAD (0.2131) for both RNNs when compared with others: a) RNN with time steps applying ADAM (0.2133), RMSPROP (0.2147) or ADADELTA (0.2147); and b) RNN with memory applying ADAM (0.2133), RMSPROP (0.4105) or ADADELTA (0.3369).

LSTM w/ Time Steps



LSTM w/ Memory

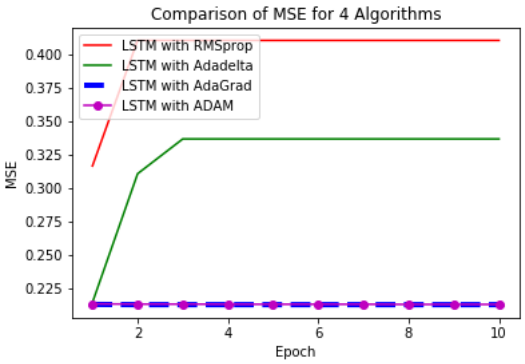


Figure 8. MSE of Two LSTM RNNs under 4 Algorithms

Case 2 – For the analysis of prediction by customer risks, we disintegrate the integrated dataset into 4 databases (MR, CR, OR & LR) and input data into the LSTM RNNs using memory, under the algorithms of ADAGRAD for prediction, as given in Figure 9.

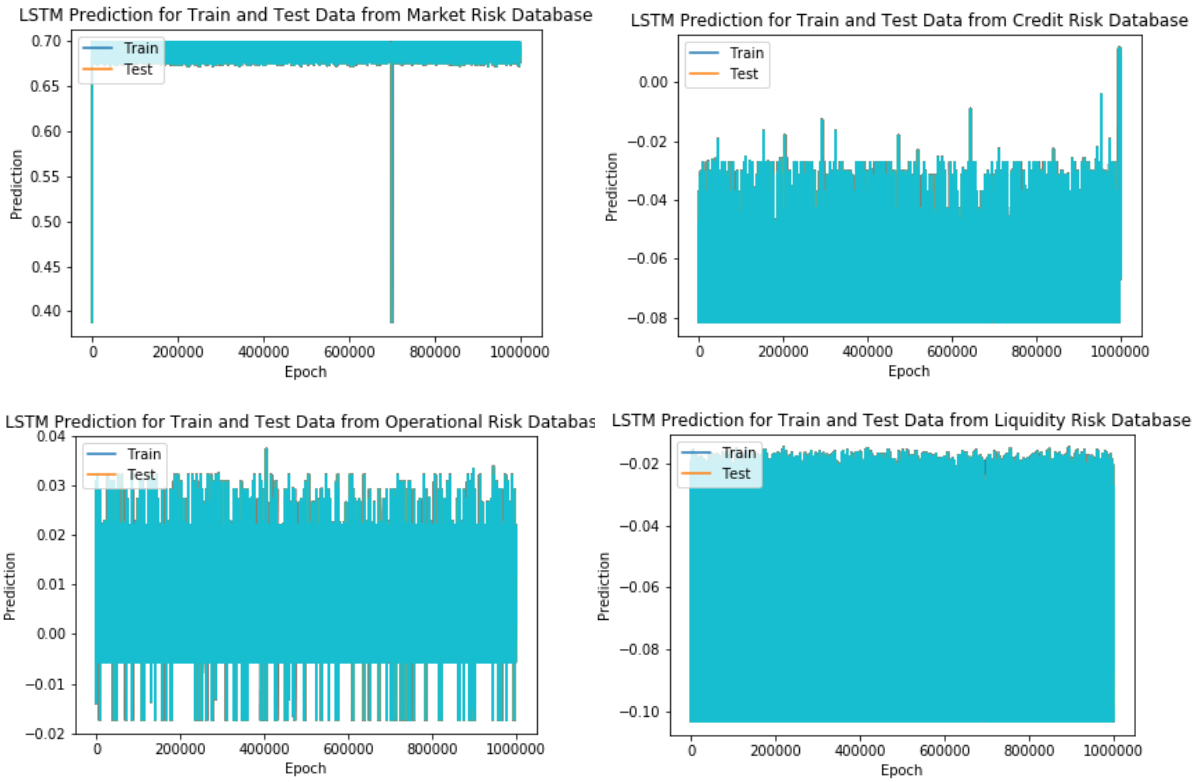


Figure 9. Prediction of LSTM RNNs for 4 Customer Risks

Utilizing ADAGRAD, we estimate precision, recall and F1-Support in Table 6. The prediction for CR and OR is high (96% and 99% respectively) but the lowest one belongs to MR (9%). LR is in between (58%). The recall and F1-Support are similar.

Table 6. Precision, Recall and F1-Support of LSTM RNNs for 4 Risks

LSTM RNN	MR	CR	OR	LR
Precision/ Recall/ F1	0.09/ 0.31/ 0.51	0.96/ 0.98/ 0.97	0.99/ 0.99/ 0.99	0.58/ 0.76/ 0.66

We also measure the error, as depicted in Figure 10. Consistently, the MSE for OR (0.0368) and CR (0.0448) is the lowest whereas the highest MSE occurs in MR (0.2133).

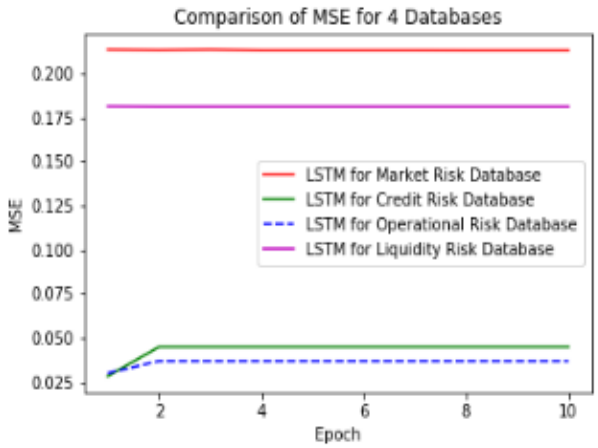


Figure 10. MSE of LSTM RNNs for 4 Customer Risks

Case 3 – For the analytics by business segments, we leverage the disintegrated databases and select retail bank ("RB"). The prediction is made in LSTM RNNs with memory under ADAGRAD, as revealed in Figure 11. To compare databases, the prediction focuses on a limited range for MR (0.33-0.365), OR (0.265-0.30) and LR (0.29-0.325) but it covers a wider range for OR (0.16-0.20) due to more quality issues as inputs generated by Python.

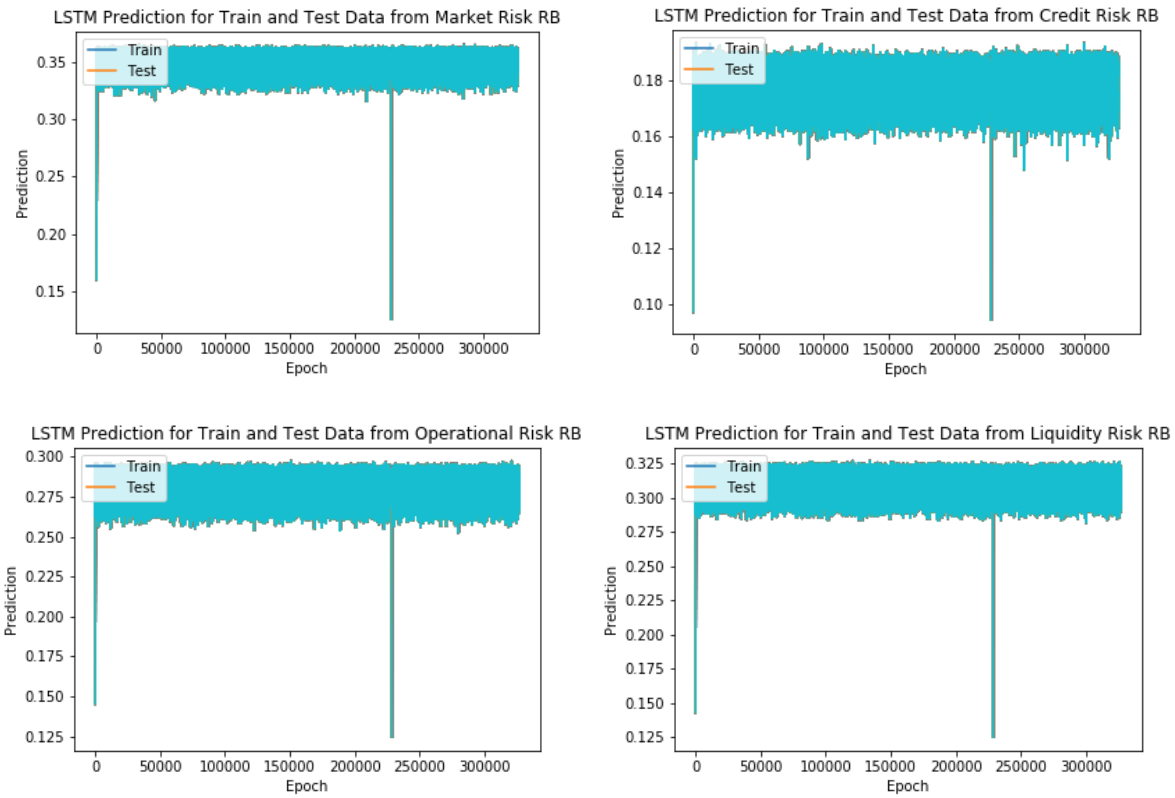


Figure 11. Prediction of the Retail Bank Segment for 4 Risk Databases

We test the LSTM RNNs in Figure 12. The validated loss for OR (0.4693) and OR (0.5939) is the lowest, similar to the MSE in Figure 13. The lowest occurs in CR (0.0177) and OR (0.0512).

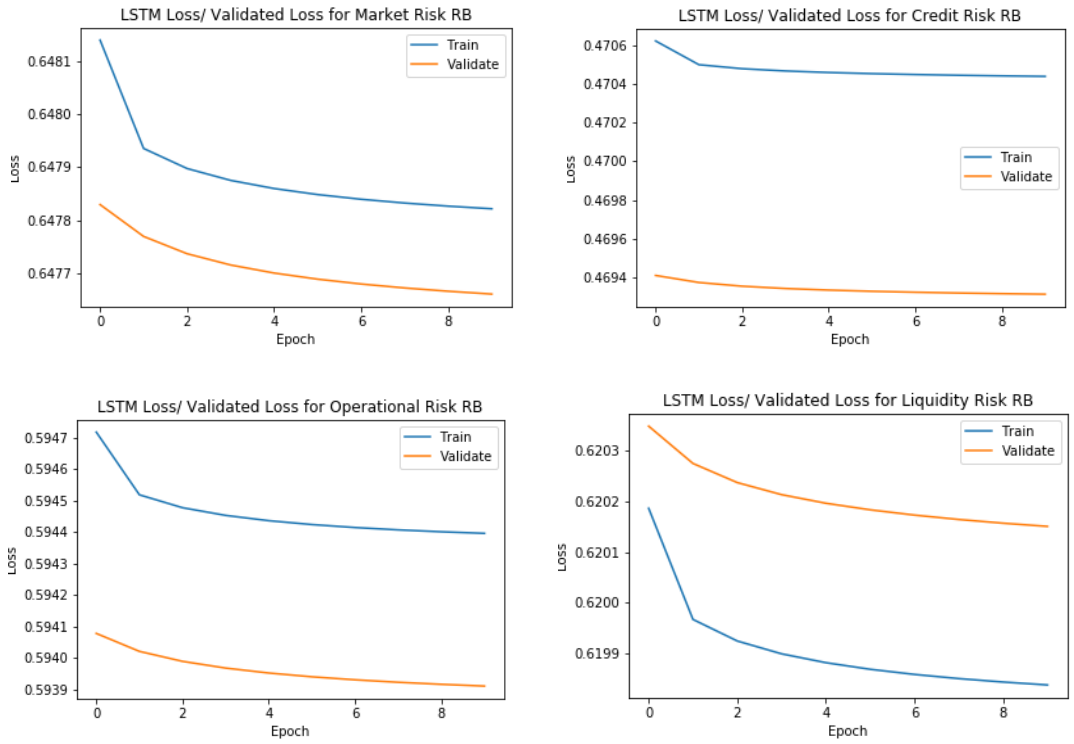


Figure 12. Loss &Validated Loss of the Retail Bank Segment for 4 Risk Databases

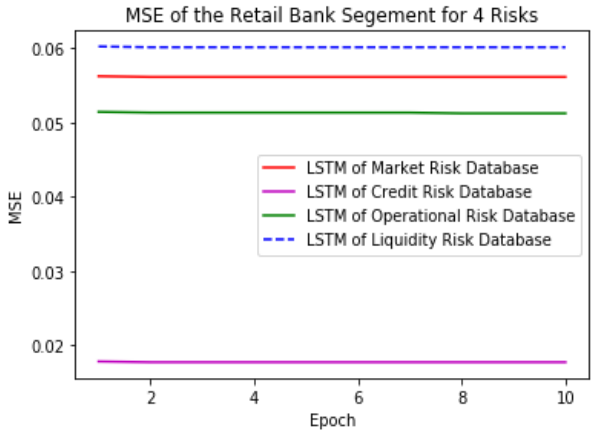


Figure 13. MSE of the Retail Bank Segment for 4 Risk Databases

Related Works

To the best of our knowledge, there is no previous work on machine learning to predict the data quality for compliance with banking requirement, CPG 235. Related works are: a) (Tavana et al. 2018) leveraged MLP and Bayesian Networks to measure and predict LR respectively. Error rate was low (8.0e-3 for GA & 1.7e-10 for LMA) while the RMSE was <0.2. Instead, ours predict the data quality of 4 risks; b) (Regina et al. 2016) used a machine learning to predict a bank credit with 23 features achieving an accuracy of 80%. But we measure the data quality with 132 features; c) (Kaya et al. 2008) classified credit with logistic regression and SVM. The accuracy was 75% but reduced to 43.5% for critical region, unlike ours experimental results; and d) (Siddayao et al. 2014) analyzed flood risk with AHP method. The importance has been defined and the hazard has been divided into 5 risks, similar to the data criticality and data quality ranking in our model.

Conclusion

After identifying the existing issues and predicting potential issues, financial institutions will understand the room for improvement in the quality of risk data in real world. This allows to remediate data as early as possible to mitigate the risk of reoccurrence. Accordingly, their analytical reports can be relied upon. But the key is the measurement of data quality in alignment with the APRA CPG 235 before analytics. This is the novelty to research field. Regardless of this, next step is to leverage machine learning to automate the data remediation for quality improvement.

References

Abdel-Nasser, M., and Mahmoud, K. 2017. "Accurate Photovoltaic Power Forecasting Models Using Deep LSTM-RNN", Neural Computing and Applications, Springer, pp. 1-14

Allen, L., and Bali, T. G. 2007. "Cyclicality in Catastrophic and Operational Risk Measurements", Journal of Banking & Finance, Volume 31, Issue 4, pp. 1191-1235

APRA. 2013. "Prudential Practice Guide CPG 235 - Managing Data Risk", Australian Prudential Regulation Authority (APRA), pp. 1-13

CFI. 2018. "Market Risk Premium", Corporate Finance Institute (CFI), pp. 2-5

Chordia, T., Roll, R., and Subrahmanyam, A. 2000. "Commonality in liquidity", Journal of Financial Economics, pp. 3-28

Crozier, R. 2017. "NAB is Building a Central Analytics Hub", IT NEWs, pp. 1-1

Dan, R., and David, S. 2010. "Risk factor contributions in portfolio credit risk models", Journal of Banking & Finance(34), pp. 336-349. http://dx.doi.org/10.1016/j.jbankfin.2009.08.002

Ergen, T., and Kozat, S. S. 2017. "Efficient Online Learning Algorithms Based on LSTM Neural Network", IEEE Transactions on Neural Networks and Learning Systems, IEEE, vol. 29, pp. 3772-3783

Fan, Y., Qian, Y., Xie, F., and Soong, F. K. 2014. "TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks", Fifteenth Annual Conference of the International Speech Communication Association, pp. 1964-1968

Frost, J. 2018. "APRA Rejected CBA Home Loan Data as Inaccurate and Incomplete", Financial Review: Business, Banking & Finance, pp. 1-1

Frydenberg, J. 2019. "Restoring Trust in Australia's Financial System", Australian Government, The Treasury, pp. 3-42

Groenendijk, M., Engelbrecht, H., and van Baardwijk, R. 2018. "Basel 4: the Way Ahead, Operational Risk, The New Standardized Approach", KPMG, pp. 3-9

Hwang, S., and Satchell, S. E. 1999. "Modelling Emerging Market Risk Premia Using Higher Moments", International Journal of Finance & Economics 4(4), pp. 271-296

IOSCO. 2018. "Recommendations for Liquidity Risk Management for Collective Investment Schemes", The Board of the International Organization of Securities Commissions (IOSCO), pp. 1-31

Kaya, M. E., Gurgen, F., Okay, N. 2008. "An Analysis of Support Vector Machines for Credit Risk Modeling", In Soares, C., Peng, Y., Meng, J., Zhou, Z.-H., and Washio, T., editors, Applications of Data Mining in E-Business and Finance: Introduction, IOS Press, pp. 27-35

KPMG. 2018. "Equity Market Risk Premium – Research Summary", KPMG, pp. 3-7

Kumar, V., Nesbit, J., and Han, K. 2005. "Rating Learning Object Quality with Distributed Bayesian Belief Networks: The Why and The How", in Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)

Migueis, M. 2018. "Is Operational Risk Regulation Forward-looking and Sensitive to Current Risks?", Board of Governors of the Federal Reserve System, FED Notes No. 2018-5-21. pp. 1-7

Moody's Analytics. 2018. "Credit Risk Calculator", Moody's Analytics, pp. 3-7

Pipino, L. L., Lee, Y. W., and Wang, R. Y. 2002. "Data quality assessment", Communications of the ACM, 45, 4, pp. 211-218

Regina, E. T., Edward, Y. B., and Gideon, E. W. 2016. "A machine learning approach for predicting bank credit worthiness", Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), ISBN: 978-1-4673-9187-0, IEEE

Shevchenko, P. V., and Wüthrich, M. V. 2006. "The Structural Modelling of Operational Risk via Bayesian inference: Combining Loss Data with Expert Opinions", Journal of Operational Risk 1(3), pp. 3-26

Siddayao, G., Valdez, S., and Fernandez, P. 2014. "Analytic Hierarchy Process (AHP) in Spatial Modeling for Floodplain Risk Assessment", International Journal of Machine Learning and Computing, 4(5), pp. 450-457

Singh, R., and Singh, K. 2010. "A descriptive classification of causes of data quality problems in data warehousing", International Journal of Computer Science Issues, 7, pp. 41-50

Tavana, M., Abtahi, A.-R., Caprio, D. D., and Poortarigh, M. 2018. "An Artificial Neural Network and Bayesian Network model for liquidity risk assessment in banking", Neurocomputing, vol. 275, pp. 2525-2554

Wong, Eric., and Cho, H. H. 2009. "A liquidity risk stress-testing framework with interaction between market and credit risks", Hong Kong Monetary Authority – Research Department, Working Paper 06/ 2009, pp. 1-33

Xu, H., and Al-Hakim, L. 2005. "Criticality of factors affecting data quality of accounting information systems", In Wang, R. Y., Pierce, E. M., Madnick. and Fisher C. W. (Eds.), Information quality, pp. 197-214. New York; M.E. Sharpe

Xu, H., Nord, J.H., Brown, N., and Nord, G.D. 2002. "Data quality issues in implementing an ERP", Industrial Management & Data Systems 102(1), pp. 47-58

Yeates, C. 2018. "Banks Dive as UBS Raises Home Loan Concerns", Sydney Morning Herald: Banking & Finance, pp. 1-2

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. 2016. "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification", in The 54th Annual Meeting of the Association for Computational Linguistics, pp. 207-213

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie. X. 2016. "Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks", in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Association for the Advancement of Artificial Intelligence, pp. 3697-3702