Predicting Book Sales Trend using Deep Learning Framework

Tan Qin Feng[1], Murphy Choy[2]
University of Stirling
Singapore

Ma Nang Laik[3]
Singapore University of Social Science
Singapore

***Abstract*—A deep learning framework like Generative Adversarial Network (GAN) has gained popularity in recent years for handling many different computer visions related problems. In this research, instead of focusing on generating the near-real images using GAN, the aim is to develop a comprehensive GAN framework for book sales ranks prediction, based on the historical sales rankings and different attributes collected from the Amazon site. Different analysis stages have been conducted in the research. In this research, a comprehensive data preprocessing is required before the modeling and evaluation. Extensive predevelopment on the data, related features selections for predicting the sales rankings, and several data transformation techniques are being applied before generating the models. Later then various models are being trained and evaluated on prediction results. In the GAN architecture, the generator network that used to generate the features is being built, and the discriminator network that used to differentiate between real and fake features is being trained before the predictions. Lastly, the regression GAN model prediction results are compared against the different neural network models like multilayer perceptron, deep belief network, convolution neural network.**

Keywords*—Generative adversarial network; deep learning framework; book sales forecasting; regression

I. Introduction

In the year 2018, the US book publishing industry achieved a net revenue of 25.82 billion USD[1]. With the invention of the internet and the creation of an online purchasing platform, finding and buying books have become so much easier and convenient. Such existence has pushed up the hardcopy and softcopy books' sales. The introduction of digital copies of books, direct publishing has made the publishing process so much simpler and accessible by both authors and readers. Not only that, the digital E-books' competitive pricing helps to stir up the book sales in the region. The book authors also enjoy getting a bigger fraction of the sales split from the self-publishing. These factors mentioned above encouraged many people to write and publish on the internet [1].

According to the Statista website, there are 45,210 writers and authors in the US and the total number of self-publication released in the US exceeded 1 million books in the year 2018[2]. With such a large number of authors available in the US and the mass amount of new book titles being published every year, the market has then become very competitive, especially for the new entry authors. This has become a challenge for them to draw attention from the mass market to their publications, and subsequently to attract readers to purchase their books.

With the improvement in computing power over the past decades, there has been increasing interest from many individuals and companies to use data science approach to predict the demand and sales across various industries. Just from the year 2009 to 2017, the most common machine learning algorithms used in new books sales forecasting are Extreme Learning Machine, K-nearest-Neighbour, Decision Tree, Artificial Neural Network, Random Forest and Service Vector Machine [2]. By mastering the demand and sales through data science, they can gain better foresight and a huge advantage in positioning their resources.

Chen found that the number of readers review is positively related to online book sales. The higher the number of reviews from different users, the better the book performed in the sales. Interestingly, her team also found that the readers' ratings on the books have no relation to the book sales [3]. In their opinion, most of the books received high ratings and dilute the trust and interest of the readers. Contradict to Chen's research, Chevalier and Mayzlin noticed that the online book sales figures have a positive relationship with respective the average stars ratings. The books with higher star ratings tend to have better sales performance than those lower ratings [4]. With such contradictive results from 2 different publications, there must be some other contributing factors that lead to the ups and downs of the sales figures.

There is another category, the time series information that frequently missed out or not available while performing logistic regression predictions or vice versa occurred as the predictions are only focused on time series data, excluding most of the other attributes needed for predictions. In this research, the past rankings of the book collected across many weeks are being included with many other relevant attributes to run the predictions. By merging the time series data with the other numerical attribute, it raises the difficulty level for

[1] https://www.statista.com/statistics/288746/global-book-market-by-region/

[2] https://www.statista.com/statistics/249036/number-of-self-published-books-in-the-us-by-format/

predicting the book rankings using the conventional algorithms available.

A. Research Questions

In this research, the study is to develop a comprehensive prediction model that is able to capture both the sales trend and ranking for both printed and digital copies from the many different attributes collected from the Amazon site. Typically, in order to solve such a problem, many conventional types of regression machine learning algorithms like linear regression, random forest, and gradient boost will come into most people's minds as the key approach. Nevertheless, in most of the companies, they do not have sufficient historical data to build a model with good accuracy for the demand and sales forecasting. Furthermore, without the previous historical records of the book sales, implementing the conventional machine learning algorithms becomes more challenging.

In the year 1991, Specht developed and introduced a general regression neural network that provides estimates of continuous variables and converges to the underlying linear or nonlinear regression surface [5]. Subsequently, many companies and researchers started to venture into deep learning algorithms for predicting and forecasting. Similarly, for this research, due to the intricacy of the data with a complex mixture of nominal, ordinal and time-series data, deep learning frameworks similar to Generative Adversarial Network (GAN) are selected in conjunction with other artificial neural networks models as forecasting techniques.

Due to inconsistency of the review studies, plus lack of features in the traditional machine learning algorithms, deep learning frameworks like GAN is able to provide better flexibilities to handle the mass amount of endogenous and exogenous variables of the books. GAN can even perform the sales trend prediction without the demand and many historical sales records. In the end, the research is to develop a comprehensive framework of deep learning that able to complete the tasks in understanding and predicting the hardcopy and softcopy books sales trends with various features.

II. Literature Review

A. On-Line-Analytical Processing and Association Rule Mining Frameworks

In the ubiquitous mobile connected society, there is a tremendous increase in the Social Network Services (SNS) data. The demand for processing mass amounts of data is rising rapidly. At the same time, the volume of collected data made it even more challenging to uncover useful and meaningful information. In order to analyse the SNS data, there are several steps need to follow through. Generally, after data is being collected, the noise in the text needs to be cleaned using Natural Language Processing (NLP) process. The detected text is then documented into matrices forms using the Latent Dirichlet Allocation (LDA) algorithm [6].

On-Line-Analytical Processing (OLAP) and Association Rule Mining (ARM) has been used as a rule-based topic trend analysis (See Fig. 1). OLAP is used to create hierarchical table formation while ARM used to extract the relevant keyword. Working hand-in-hand, they are used to identify previously unknown information and special events. Park and his team (2017) used OLAP and ARM to analyse the social trend and identify similar discussion topics from different users and insights. They showed the feasibility of a combination of the two different data mining techniques [6]. However, challenges still remained for a better understanding of topic trends. Since the frequencies of each topic are classified as measure values in the fact table, in order to handle other types of measured values such as the relative ratio of topics, structure and unstructured data, another deep learning framework is still required.

B. Knowledge base Neural Network Frameworks

In the rise of intensified competition to capture readers' attention, it has then become very important to understand and model online popularity dynamics. Many researchers have been exploring feature-based methods such as random forest and regressions in tackling the task. Since the data is rich in contents and context, Dou and his team used heuristically link online items with existing knowledge base entities to improve the popularity prediction [7]. Fig. 2 shows the schematic diagram of the proposed model. The team has utilized the time series and context data collected in order to generate a robust prediction model.

There are 3 key important issues that need to be considered for the context information popularity prediction. They are types of general contexts used, unified and compact way of representation, and lastly the integration and utilization of the context. To address the issues, a knowledge base neural network is embedded in the Long-Short Term Memory (LSTM) networks for predictions. The prediction gained further improvement when Dou and his team integrated knowledge base neighbours that are used to help to group similar popularity dynamics [7]. The experiment results showed that both the popularity dynamics of the knowledge base neighbors and embedding of the target item improve the prediction results. However, not all the entities can find corresponding knowledge base entries, other methods can be explored to enhance the prediction performance of nonlinked items.

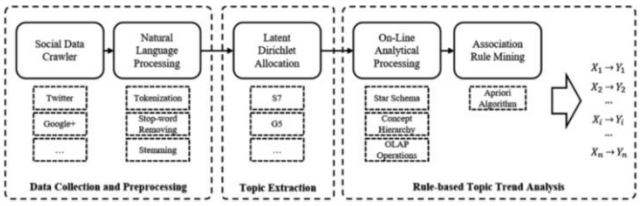

Fig. 1. The Overall Architecture of the Proposed Method [6].

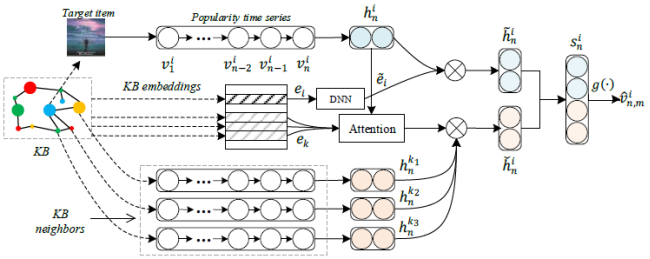

Fig. 2. The Overall Schematic Diagram of the Proposed Model [7].

C. Neural Graph Collaborative Filtering Frameworks

Collaborative filtering (CF) has been widely used in estimating user adoption rate on an item based on the pass interaction behaviours. There are two key components in the learnable CF models. Learning vector representations (aka. Embeddings) that transform the users and items to vectorize representation. The other component is the interaction modeling or matrix factorization (MF) that reconstructs the historical interactions on the embeddings. Due to the lack of explicit encoding of the collaborative signal in the CF, Wang's team developed Neural Graph Collaborative Filtering (NGCF) to make up for the deficiency of suboptimal embeddings [8]. See Fig. 3 for the architecture of the NGCF model developed by Wang's and his team.

By adding an embedding propagation layer, the collaborative signals between the users and items can be harvested for analysis. However, the NGCF still needs further improvement by adding connectivities of different items' orders. The attention mechanism to learn variable weights for neigbours during the embedding propagation also can be enhanced by introducing other models like adversarial learning [8].

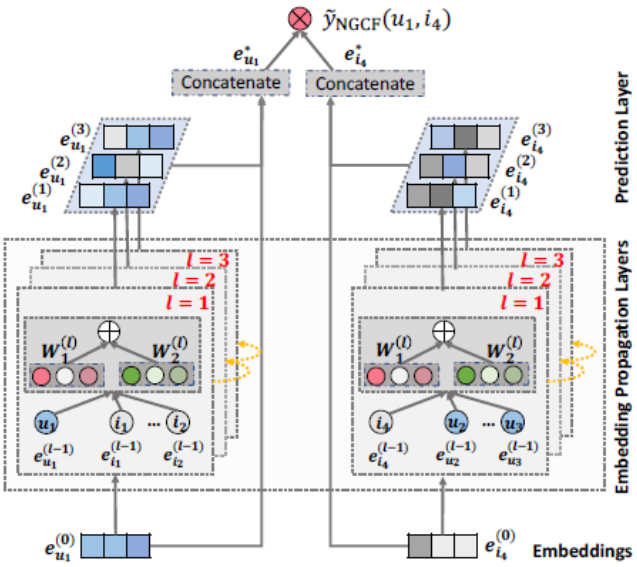

Fig. 3. An Illustration of NGCF Model Architecture (the Arrowed Lines Present the Flow of Information). The Representations of useru1 (left) and Item i4 (Right) are Refined with Multiple Embedding Propagation Layers, whose Outputs are Concatenated to make the Final Prediction [8].

III. Related Work

Recently, there is a lot of attention to the Generative Adversarial Network (GAN). GAN has been introduced by Goodfellow and his team to simultaneously training both the generator model and the discriminative model. GAN has been widely used on a large structure like images with multi-dimension and big output space [9]. Little attention is being drawn to use GAN in solving a single dimension data like classification and regression related problems that only generate small output space. Until recent, Aggarwal and his team started to introduce Conditional Generative Adversarial Network (CGAN) [10] as another comparative model in regression prediction. On the other side, Autoregressive Integrated Moving Average (ARIMA) is one of the popular models when comes to the time series prediction. In Zhang's research, he and his team have selected the GAN model with Long-Short Term Memory (LSTM) network as the generator and Multi-Layer Perceptron (MLP) as the discriminator for predicting the stock price [11].

Studies have been made on using GAN on regression and time series prediction. Both types of research using GAN for prediction are yielding positive results compared against many other deep learning models. As for our research, the data we collected is made up of a mixture of both forms of information. Hence, the GAN framework is being selected and used predicting the book rankings based on all the books' features presented and the past rankings collected.

IV. Using the Template

The methodology in data science requires a structural system of methods to derive particular models targeting a specified area or study. Cross-Industry Standard Process for Data mining (CRISP-DM) methodology is being applied to track and monitor the different milestones of the project. The 6 keys stages are business understand, data understanding, data preprocessing, modeling, evaluation, and deployment [12]. With the CRISP-DM functional template, this will ensure proper procedures being follow-through during the research in order to generate good functional models in predicting the book sales trend.

There are three datasets being selected and used in the research. All the thrrr datasets that were used in this research are considered as secondary data. They are easily accessible from the open dataset site like Kaggle. The first dataset are originally gathered from the Amazon sites. The first dataset contains "ASIN" (a 10 characters long unique Amazon identifier), and other key attributes like the title, authors and publishers of the books. The group and format contained in the dataset helped to distinguish physical and digital versions of the particular ASIN code. The second dataset is focused on Kindle edition. Besides the basic data, the Kindle dataset has the rating, price, number of pages for the books, and some other text attributes which are the languages, lending, customer reviews, short descriptions of the books being published and etc. that the first set lack off.

From both different datasets, they consist of three different types of data. They are made up of quantitative and qualitative data that comprised of numerical, cardinal and text format. Lastly, the third set is made up of collective rankings of a particular ASIN from 1st January 2017 until 29 June 2018. There are totals of 118,200 JSON files being collected. The captured rankings from each file are named in ASIN code, relative to the first dataset. Each individual JSON file represents the book's ranking across the 77 weeks, collected as frequent as an hour to 24 hours interval and the rankings recorded are stamped in binary date-time format. There are some books without ranking initially during the early weeks as they probably not published yet during that period.

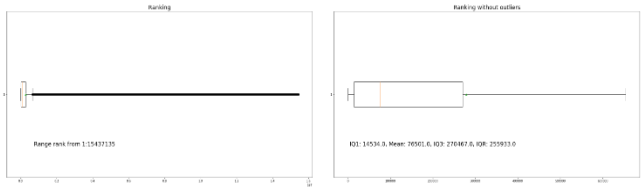

Fig. 4. Boxplots of the Book Rankings with and without Outliers.

Fig. 4 shows the boxplot for all the book rankings collected from the combined dataset. The figure on the left shows the boxplot with the outliers while the figure on the left is the boxplot exclude the outliers. The books' ranks range from the best rank of 1 until the worse rank of over 15 million rankings during the collection period over 77 weeks. From the plot below, most of the books' rankings fall below 700,000 with the average ranking at 76,501 which is only about one-tenth of the majority. Therefore, in our dataset, we noticed that most of the books are performing rather well in their rankings.

V. Setup and Preprocessing

Preprocessing the collected data is one of the important stages before constructing the model. To generate a good model, the quality of data needs to be considered. Not all the single data collected is instantaneously suitable to train and to build the model. Whatever input fits into the model will greatly impact the performance of the deep learning framework and later further affect the output. Hence, the data that is going to feed into the modeling process has to be carefully selected, the Not a Number (commonly appear as NaN) has to be replaced and the entry errors are required to be removed from the list before allowing them to the built the algorithm.

A. Combine and Setup

Firstly, the JSON files are being processed by merging all the individual files into one comma-separated value (CSV) files for processing in the later stage. The binary date format in the JSON files are being converted to the readable date-time format as well as the ranking value captured on that particular date and time are stored together in one single file. After the conversion of the files, we proceed to extract the weekly highest ranking and the last ranking achieved for all the books as the output value for training later. As for the other two datasets, they contain rows of books and columns of attributes collected for the book. Removing of missing information and entry errors of the datasets are performed to ensure that the wrong information is not being selected during the modeling process. Lastly, using the unique identifier–the ASIN from all three sets of files, we merged the entire data from the three files into one big dataset.

The 77 weekly highest rankings for the books extracted from the individual lists are also being transposed and inserted as 77 separate columns. The last of the week ranking will then become the targeted sales ranking for the entire research. By merging all the disconnected data into one single set, it will help us to have a better understanding of the relationships between all the variables contents, and enable an effective model building later in the process. Fig. 5 shows the overall rankings being captured across 77 weeks for 2 different books. Fig. 6 shows the mined result from weekly top ranking for the same 2 books. Compare Fig. 5 and Fig. 6, the original graphs' patterns of the overall books' rankings are kept the same even after extracting the top weekly ranking. This will greatly help to bring consistency and uniformity for the model building on the other different books.

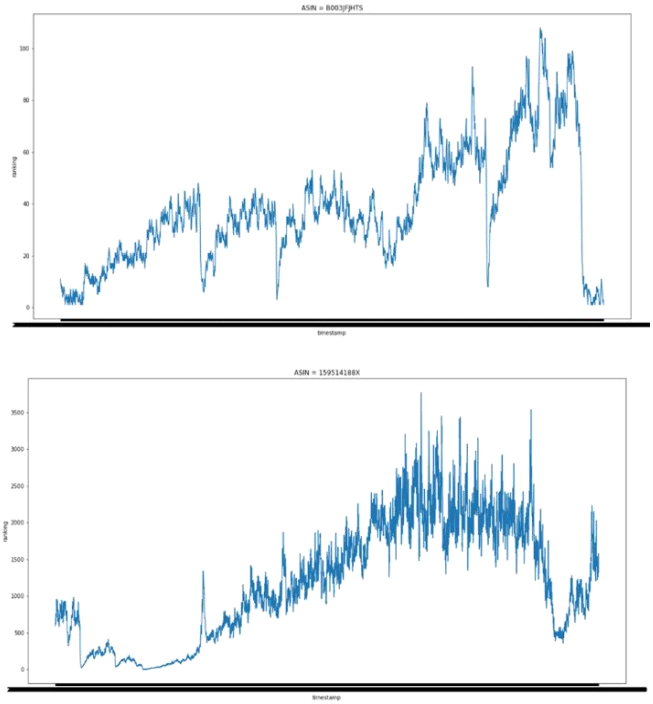

Fig. 5. Snapshots of 2 Books Full Ranking Across 77 Weeks.

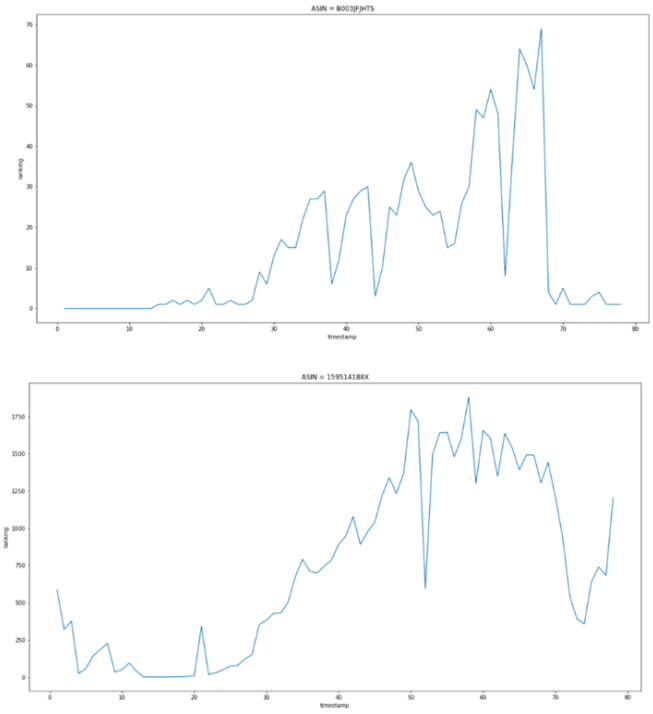

Fig. 6. Snapshots of 2 Books Weekly Top Ranking Across 77 Weeks.

B. Clean and Transform

Once all the data is contained into 1 big set, the data underwent another round of preprocessing by removing duplicates, and all other columns that are not required during the modeling process. Nominal and categorical columns are also being replaced with numerical values for subsequent analysis. As an example, in the 'Lending' feature, the 'Enabled' is replaced with '1' and 'Not Enabled' is replaced with '0' and the four different 'Format' of the book published is replaced with an integer '1~4'.

After the comprehensive constructions and setup are completed, all the preparation works accomplished in the preceding process is to support the main goal, which is to get the data ready for analysis and model building. After merging, the single dataset comprised 108 different columns, and each column representing a parameter. However, for building models and fitting into the mathematical regression, not all 108 features are useful and applicable. Over at this stage, it is necessary to execute another level of data cleaning again, just to make sure the parameters selected are truly meant for training the model. Those unique features that do not belong to any categorical structure will be dropped from the dataset. As an example, the title, authors, publishers, and URL of the books are being removed from the study. In the end, once the cleaning and removing of the unwanted columns are completed, we are left with 90 useful parameters that can be rescaled and transformed for modeling.

To execute the deep learning framework modeling process, the values contained in the dataset need to be transformed into the accepted format. All the features data required to be rescaled, using a min-max scaling formula. It is important to scale the values in the dataset as it allows the relative differences among the values to be treated as equal terms. Plus, it helps to increase proficiency in the arithmetic operations. Especially during the model building process, the transformed values can be used unambiguously by the deep learning framework [13]. Each individual feature is transformed using the formula (see Equation (1)) below, with the range from zero to one.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

where x is the original value and the x' is the new scaled value. In this research, total ranking values were scaled by multiplying by 0.0000000775 and adding 0.

C. Split and Divide

In order to better understand the model performance, it is important to split the data into two different training and validation sets. The prediction results from the validation set allow the user to access the model accuracy after training the model [14]. After all the merging, cleaning and converting on the values in every column, we obtained 1932 rows and 90 columns of good information that can be used for modeling. For the training and validation of the models, we split the total into a ratio of 80:20. We have 1546 sets for training the model and 386 sets to be used during the validation process. From the 1546 sets, we need to have another separate train and test set when building the model. Hence, from the Train_Test_Split function in Python library, we set aside another 30% from the 1546 sets as test dataset. In the end, we have 1082 sets for training; 464 sets for testing and 386 sets keep aside for evaluation after the modeling. To achieve consistency during the entire study, the random state for all the settings is set to state zero.

Cross validate is another commonly use data splitting method in the modeling process. The sample-set is randomly divided into k different equal size subsamples where k can be any integer number and often called a number of folds. The k-1 subsamples are used to train the model and the remaining one sample is used to validate the model. The entire process is then repeated with k numbers of times and performance on each fold is recorded for evaluation [14]. In our research, we divided our samples into k=5 different folds in the modeling process. And the highest, lowest and average scores of the cross-validate results are recorded for further evaluation later in the process. Fig. 7 illustrates the 5 fold cross-validation technique.

D. Correlation

Continue after the preprocessing, we followed by performing a Pearson correlation on all the features in the well-cleaned dataset. This is to evaluate the statistical relationship between the variables. The Pearson correlation coefficient r [15] tells the strength and direction of the linear relationship between the 2 variables. The correlation between the two variables can be denoted as r_{xy} and computed as Equation (2):

$$r_{xy} = \frac{cov(x,y)}{\sqrt{var(x)} * \sqrt{var(y)}} \quad (2)$$

where the cov is the sample covariance of x and y, var is the sample variance. The two values gain a stronger positive relationship as the rxy moves closer to +1. As the x value increases, the y value will also increase along with x for a positive relationship, vice versa, as the results move closer to -1. When the r value moves toward the direction of 0, the relationship between both values grows weaker. Lastly, if r=0, both values show no relationship at all.

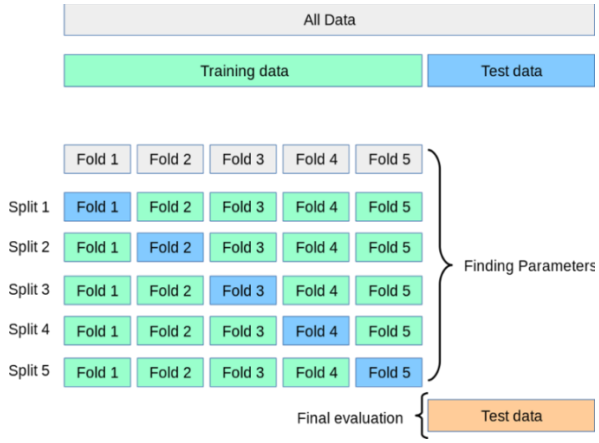

Fig. 7. Illustration of Cross-Validation Technique[3].

[3] https://scikit-learn.org/stable/modules/cross_validation.html

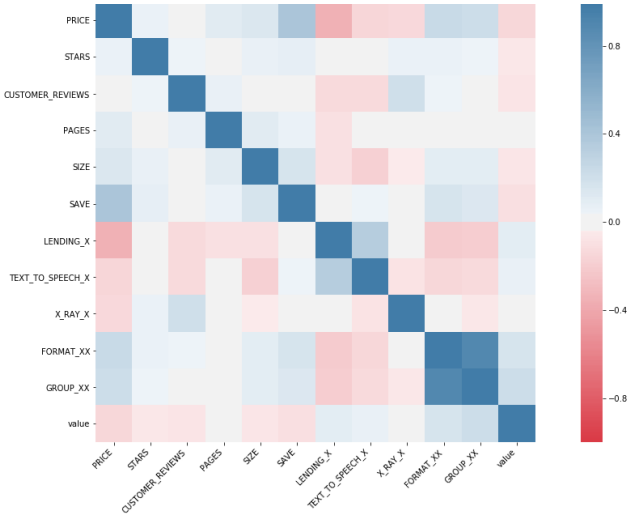

Fig. 8. Snapshot of Pearson's Correlation Heatmap with Values.

In the correlation research, all the top weekly rankings are being excluded from the study. The aim is to study the book's features in response to the final ranking value. From Fig. 8 below, we can see that all the features do not possess a strong relationship with the ranking. There is also a mixture of positive and negative correlations with the last ranking value. Important note for this research, the correlation heatmap shows that the price has a weak negative correlation with the ranking value. Though, the ranking of the book sales is inverse with the numerical number. The lower the numerical value appears in the ranking, the better the book performs. Hence, a clear example is as the price of the book goes up, the book ranking will likely drop because the ranking number grows bigger.

VI. Modeling

There are 3 major types of data analytics for businesses and researchers in understanding and deriving useful information from data. They are descriptive analytics, predictive analytics, and prescriptive analytics. Depending on the values and information that each individual intends to extract out from the data, it requires a different set of analytical techniques. For our research, the focus is to predict the book sales ranking base on the book features and historical ranking collected over a period of time. Predictive analytics [16] is more suitable for our research. It involves a variety of statistical techniques to make predictions about the future. Therefore, predictive models in machine learning are to be selected and applied to generate the desired outcome that fulfills the purpose of the entire study.

In the data mining cycle, the modeling phase is the heart of the process. Just like the heart, it pumps and supplies blood with nutrients to the entire body, the created and selected model is vital to assist businesses and researches in providing accurate and desired results. During the modeling phase, various modeling techniques are chosen and trained once the data, features and models' parameters are properly setups. Generated models are tested, assessed and possibly revised again on parameter settings in order to obtain a perfect outcome. In this paper, Multilayer Perceptron (MLP), Deep Belief Network, Single and two-dimensional Convolution Neural Networks (CNN) are the few deep learning algorithms being selected as a study to compare with the GAN framework. These few artificial neural network architectures comprise of many nodes and several networks connected by one layer with another in sequence to produce the desired results. Research has been conducted using the mentioned neural networks above to understand the performance of each different type of deep learning algorithms in predicting the book sales ranking.

A. Multilayer Perceptron (MLP)

The first neural network algorithm that we built is the Multilayer Perceptron (MLP). MLP is one of the most common neural networks and its architecture is also the fundamental architecture for the majority of the neural networks. It is also simple to build and widely used by many researchers. It comprises an input layer and an output layer. In between, it can have many hidden layers connected between the input and output. In the layer itself, it can have one or more artificial neurons called perceptrons. Each perceptron carries weight with activation function to produce a value for the next layer. The output from each node can be represented as Equation (3):

$$h_{w,b}(x) = g(\boldsymbol{w}.\boldsymbol{x} + b) \quad (3)$$

where $g(x)$ is the activation function, w is the weight leading to the node and b as the bias [17]. The multilayer perceptron architecture is shown in Fig. 9.

In this study, we constructed a 4 layers MLP. The first layer is the input layer. As we have a total of 90 different columns in our dataset with 89 variables and 1 output, the input dimension for the first layer in the MLP is set to 89 with 360 nodes, a rectified linear unit (ReLU) as the activation function. Subsequently, the other 2 hidden layers are set with 540 nodes and 180 nodes respectively with the same ReLU activation function. The fourth output layer is set to 1 node that will be the output results from the algorithm. As for the output layer, the activation function is set as linear instead as the results will be in a linear regression form. Adam optimizer with the default learning rate of 0.001 is selected when constructing the MLP compiler and train with 1000 epochs.

B. Deep Belief Network (DBN)

Similarly, the deep belief network also comprises of 3 different layers. First is the visible layers where the input values are inserted. The next hidden layers are built with Restricted Boltzmann Machines (RBMs) [18] and last is the single output layer that can be in the form of classifications or mathematical regression. The mutual graph of the visible units that represent observations are connected to binary, stochastic hidden units using undirected weighted connections are called RBMs. They are restricted because there are no visible or hidden connections between them. The model gets refined and improved as the hidden layers distribution of the model keeps replacing whenever a better model that is learned by treating the hidden activity vectors produced from the training data as the training data for another RBM. The RBMs have an efficient training procedure which makes them suitable as building blocks for DBNs.

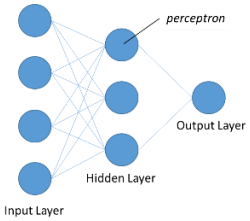

Fig. 9. Multilayer Perceptron Architecture.

In the DBN structure, the θ is the weight of the model, h is the vector of the hidden layer with the distribution of p(h|θ) and the probability vector as p(v). The formula is written as Equation (4) below [18]:

$$p(v) = \sum_h p(h|\theta)\, p(v|h,\theta) \quad (4)$$

Fig. 10 shows the deep belief network architecture. For our DBN model, we created it similar to the MLP with 4 layers. There are 1 visible layer, 2 hidden layers and 1 final layer as the output layer. The number of nodes available for each layer is set to the same number of nodes as the MLP in the ratio of 360:540:180 and the output layer is activated with a linear regression model. In all the RBM layers' settings, the learning rate is set to 0.1, 5 iterations over the training dataset during the training process and 10 mini-batch sizes.

C. Convolutional Neural Network (CNN)

As the data size gets larger and more dimensions are being introduced to the dataset, especially in the larger images and video contents, the classic neural networks take up a lot of memory space and require very huge computing power to process them. The Convolutional Neural Network (CNN) [19] is then introduced to handle the bigger appetite on data management and data analysis. From the word itself, the neural network uses the convolution technique instead of the matrix multiplication on its layer. The kernel that is much smaller than the input size is put through the activation function to form the output feature map. Pooling function is another feature in the CNN that use to down-sampling the input in order to further increase the receptive field of the outputs. Single or multiple iterations of the convolution process are performed until the parameter gets flatten into a single dimension layer. Then the following process to get the output is similar to MLP once it is flattened [20]. Compare to the traditional neural network consist of 3 layers, the input, the output, and the hidden layer; the algorithm's parameters got reduced and the complexity got simplified by adding the convolution layer and sub-sampling layer [21].

As a whole, the output for CNN is (see Equation (5)):

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (5)$$

where *Mj* is the selection of the input parameters. *i, j, k, l* representing the input map, output map, kernel size, and the convolution layer. Lastly, figure *b* is the additive bias from each output map [19]. The convolution neural network framework is displayed in Fig. 11.

Two different types of convolution neural network (CNN) algorithms have been produced in predicting the book sales rankings. We created a single dimension CNN (1DCNN) and two dimensions CNN (2DCNN) models in this research. For 1DCNN, it is used for input signal patterns like voice and time-series data; and for 2DCNN, it is generally meant to process input signal like images. As for our dataset, the structure is closer to 1 dimension input with a single row and a mixture of numerical and time-series information. For 1DCNN input, we reshape the train and test set by introducing a single channel filter to the dataset with the shape of (row, columns, channel=1). Nevertheless, we can still perform 2DCNN on our dataset by reshaping the data into a 2-dimensional array. We introduced the second dimension and a single channel filter as well to the same train and test dataset into the shape of (row, columns, additional dimension=1, channel=1).

For both 1D and 2D CNN setup, we have two convolution layers, and batch normalization function right after the convolve layers to regulate and normalize the input layers[4]. After the convolution processes, we flatten the signal and add a fully connected layer before the signal is passed to the output. Between each layer, we introduced a 50% dropout regularization technique to prevent overfitting in the neural networks [22]. As the research is focusing on regression output, the linear activation function with $y = ax$ can be used for continuous output. Hence all the layers from input until the output are set with a linear activation function. The filter size for all layers except the last output layers is set to 90 for 1DCNN and 2DCNN. The second input that is set to the convolution layer in the neural network is the kernel size. For our 1DCNN model, the convolution layer kernel size is set to 2 while the 2DCNN model kernel size is set to 1. Both algorithms are trained with 1000 epochs. Adam optimizer with the default learning rate of 0.001 is selected when constructing the 1DCNN and 2DCNN compilers.

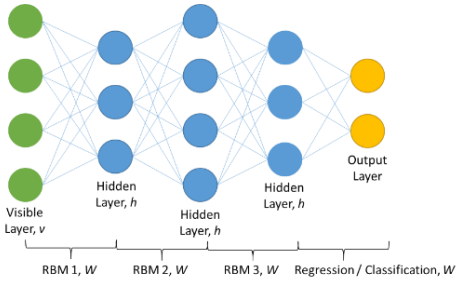

Fig. 10. Deep Belief Network Architecture.

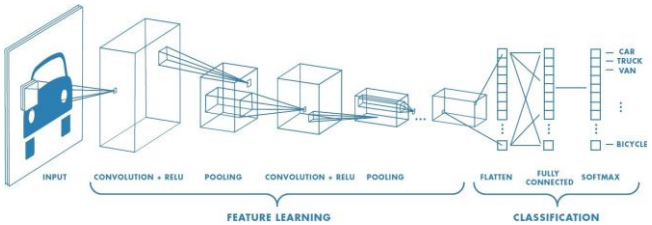

Fig. 11. Convolution Neural Network Frameworks[5].

[4] https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c

[5] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

D. Generative Adversarial Network (GAN)

GAN is an interesting and unique deep learning framework. It has certainly gained a lot of attention and popularity. GAN functions by integrating two different neural networks, the Generator (G), and the Discriminator (D) to compete and work together simultaneously, just like the zero-sum game from the game theory [23]. In order to cheat the discriminator, the Generator's role is to generate data that mimic the real data by taking in random noise. On the other hand, the Discriminator's role is to distinguish real and fake data, input from both real datasets and generator sets. Both algorithms are trained together until it reaches a stage that the generator is capable of generating the fake data that the discriminator unable to classify as fake input [11]. In summary, the Discriminator function is to maximize the probability of identifying correct input while the Generator is continuously trained to minimize the probability of letting the Discriminator identified as the generated output as fake input [20].

Fundamentally, the generator and the discriminator are running on two different neural network algorithms as compared to the conventional deep learning framework that performed base on a single neural network. The GAN algorithm starts by defining a prior on the input noise variables *pz(z)* that will be taken in by the generator that represented as neural network *G(z; θg).* The *G* is a differentiable function represented by a neural network with parameters *θg*. As the second definition for the discriminator neural network *D(x; θd),* that provides a singular scalar score The *D(x)* denoted the probability from the sample data *x*. The *D* then trains to maximize the probability of assigning the correct label taking the input from both sample data and generator's output. Whilst, the *G* is trained to minimize the rejection from *log(1-D(G(z)))* [9]. The GAN's model equation and diagram are shown in Equation (6) and Fig. 12.

$$min_G max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{Z\sim P_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (6)$$

For typical GAN would need to be trained with ten to hundreds of thousand iterations to get the optimum results. Conditional GAN or known as CGAN framework is used to assist in the entire GAN prediction process. Just like the GAN, CGAN also has both the generator and discriminator networks. On top of the fundamental *x* and the noise *z*, both generator and discriminator are conditioned with the third variable *y*. The *y* is the information that can be in any form like classification labels or continuous values. It acts as an additional input to both the generator and discriminator for conditioning purposes. *y* joint in the *z* together as an input *p(z|y)* for the generator and presented together with the x as an input *p(x|y)* for the discriminator. This helps to provide boundaries for the expected outputs and speed up the entire training process in the GAN network by giving the generator and discriminator this direction [24]. To illustrate further, the equation for the CGAN and framework are as below (see Equation (7) and Fig. 13):

$$min_G max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x|y)] + E_{Z\sim P_z(z)}\left[\log\left(1 - D\left(G(z|y)\right)\right)\right] \quad (7)$$

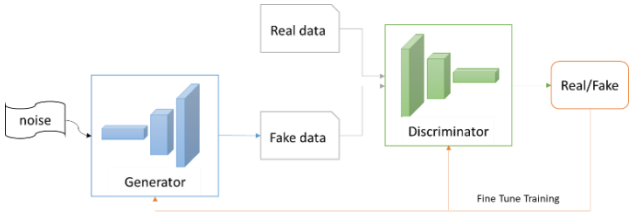

Fig. 12. General Generative Adversarial Network Framework.

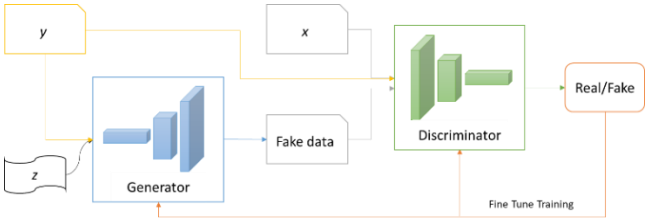

Fig. 13. Conditional General Generative Adversarial Network Framework.

As for our GAN framework, after receiving the random noise input, the generator requires to generate fake data that has the same row and column with the real data. We built a generator network consist of five layers of MLP framework that will take it the noise z and y values to generate out floats in the array format of 89 columns and 1 rows. From the input layer until the final output layer of the generator network, the number of nodes ratio with respect to the number of attributes of the dataset is set to 1:3:2:1:1 in sequence. At every neural network layer, LeakyReLU activation function is added for weight rectification on the nodes. The constant multiplier, α with the value of 0.2 is being set on every LeakyReLU function in the generator network. Batch normalization function is also being inserted after the activation function with the momentum value of 0.8 helps to reduce the noise in the gradient. *tanh* activation function is selected for the last output layer in the generator network. The generated output format is then reshaped to make sure the result is identical to the real input *x* that will be feed to the discriminator network later for identification.

In the discriminator network of our GAN design, it is tasked to handle the real and fake *x* input from both real data and generated data by the generator network. Five layers of MLP framework is also being modeled in the discriminator network to handle data array with 89 columns, 1 row. Similarly, the node ratio for the discriminator network at every layer is also set to 1:3:2:1 accordingly except the last output layer is just a single node. LeakyReLU activation functions with α of 0.2 are being inserted between the layers. To reduce the overfitting, a 50% dropout rate is set for every discriminator layer. Sigmoid activation is set to the single node at the last output layer for true false identification for the discriminator.

Once set, the GAN is trained with 5000 iterations on the training dataset and the discriminator weight is saved for the prediction later. While training the GAN, we included the condition value y, which is the final ranking value of the training dataset to the generator network and discriminator network in order to hasten and smoothen the training process. Subsequently, the discriminator weight is load again and train

with another 1000 epochs for prediction of the book sales ranking.

VII. Evaluation

To understand the reliability of the models built based on the training dataset, the models need to be further evaluated by feeding them with the testing dataset. There are many different types of evaluation techniques that can be applied to understand the performance of the models. Different types of models require specific evaluation techniques to measure their performance and reliability base on their respective outputs. For example, the confusion matrix is well known to evaluate classification type of output while the mean square error and mean absolute error is commonly used in the regression. Therefore, applying the correct evaluation metrics, analysts and business owners can understand how the models behaved before selecting a suitable model for real-world deployment. For our predictive modeling research, the Mean Absolute Error (MAE), the Mean Square Error (MSE) and Root Mean Squared Error (RMSE) are selected to evaluate the deep learning models computed books' rankings against the collected rankings.

A. Mean Absolute Error (MAE)

MAE is a quality measurement metric that widely used in regression evaluation. It is used to measure the absolute difference between the actual values and the estimated values. The MAE formula is defined as Equation (8):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (8)$$

where n is the number of the sample size, y_i is the original observed value and $\hat{y}_i$ is the predicted value. In this context, the n is the size of the testing and predicting dataset and y is the final captured book's ranking across the study. The model's predicted ranking is represented by the $\hat{y}$. When using the MAE for evaluation, the smaller the number, the better the predicted values as they are closer to the expected values.

B. Mean Square Error (MSE) and Root Mean Squared Error (RMSE)

Similar to MAE, MSE is another type of quality measurement metric that popular in the regression evaluation as well. Instead of absolute difference, it measures the average squared distance between the actual values and the estimated values. However, both MAE and MSE have a slight difference in the meaning of the value calculated. The MAE ignores the direction or the negative values from the calculation. Whereas the MSE squared the differences between observed and expected value. Hence, the MSE carries more weight to a bigger loss in the calculation, in which larger errors are particularly undesirable. MSE formula is defined as Equation (9):

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (9)$$

As for the RMSE formula, it is just the square root for the MSE. For MSE and RMSE measurements, similar to MAE scores, the smaller the MSE value, the better the predictive model generates results.

C. Results Comparison

Besides the 30% testing dataset that split out from the training set, we have another 386 prediction dataset that is not part of any training and testing set which is meant for evaluating the algorithms on untapped datasets outside the training and testing sets. The prediction dataset was randomly separated during the preprocessing is to ensure that none of the data selected during the modeling phase is being recycled again for training. This is to show the reliability and performance of the model in predicting the unknown dataset. The tables below shows the MAE, MSE and RMSE scores for the 5 different types of deep learning networks generated on predicting the book sales ranking against the original value. The percentage scores in both tables show the delta Δ of improvement for the models comparing the test results and prediction results. The formula of the Δ improvement is as Equation (10):

$$\Delta\% = \frac{y_{test} - y_{predict}}{y_{test}} \quad (10)$$

In the first glance from Table I, the smallest figure tabulated starts with 5 digits and the largest integer can go larger than 350,000. It seems to have a rather big MAE score in the books' rankings predictions. However, when we compared with the books' rankings ranges in 10 of millions, the worst performance neural network, 1DCNN scores around 350,000 is only about 3.5% from the total ranking value. Therefore, the evaluation scores are considered acceptable. From the 5 neural networks, MLP overall scores the best MAE for predicting the book rankings for both test and predict dataset prepared. Whereas the GAN frameworks that leverage on MLP architecture for the generator and discriminator networks perform second best and it is just a few thousand scores off from the MLP. Nevertheless, when comes to the comparison between the test set and predicted set, the MLP, 2DCNN, and GAN saw a positive difference. Among the 3, the GAN framework has the best improvement of about 5.5%. It is more than 5% better as compared to MLP which does not see much improvement between the test set and predict set. If more datasets and training can be provided to GAN, it can definitely generate better predictions for the books' rankings.

Table II shows the MSE and RMSE results scored by each neural network in predicting the books' ranking for test and predict dataset. Due to the MSE squared function, plus the large ranking values and the big ranking ranges possessed within the dataset, the scores grow so huge that the worst performance 1DCNN results reach beyond 11 digits. Hence the RMSE helps to reduce the dimension so that it is much easier to read and decipher. Among all five different frameworks, GAN has the lowest RMSE values for both test and predict sets. Within the GAN framework, the scores also reduced from test RMSE to predict RMSE with an overall improvement of 14.25%. Although it seems 2DCNN ranked second at the improvement scores, the 2DCNN RMSE values for both the test and predict are more than double the value of the GAN RMSE scores. MLP algorithm shows the best reduction of the RMSE value at 16.77% from the test to predict the dataset. Nonetheless, our GAN generator and discriminator networks are using the MLP framework to perform hand in hand together for generating the prediction on

the books' ranking. Similarly, as the MAE results, if we can perform more iterations of training to the GAN framework, we can see even better RMSE scores as well.

TABLE. I. MEAN ABSOLUTE ERROR (MAE) RESULTS

Neural Network	Test MAE	Predict MAE	Δ Improved
MLP	**80836**	**80815**	0.03%
DBN	94681	109936	-16.11%
1DCNN	352150	355656	-1.00%
2DCNN	181876	174085	4.28%
GAN	89723	84787	**5.50%**

TABLE. II. MEAN SQUARED ERROR (MSE) AND ROOT MEAN SQUARED ERROR (RMSE) RESULTS

Neural Network	Test MSE	Test RMSE	Predict MSE	Predict RMSE	Δ Improved
MLP	2.73e10	165108	1.89e10	137415	**16.77%**
DBN	2.25e10	149909	4.08e10	201872	-34.66%
1DCNN	1.67e11	409053	1.73e11	416005	-1.70%
2DCNN	9.02e10	300352	6.44e10	253713	15.53%
GAN	1.98e10	**140611**	1.45e10	**120576**	14.25%

MAE and RMSE are often used together as accuracy indicators for continuous variables. By having both the indicators tabulated together, we can derive another level of information. From the definition, RMSE will never be smaller than MAE, as RMSE ≥ MAE. Both values can be in the range from 0 to ∞. By subtracting RMSE and MAE, we are able to understand the variation of errors in the forecasted results. The larger the remaining value from the subtraction, the greater the variance the individual error can be found in the sample set. When both errors are having the same magnitude, then RMSE is equal to MAE [25], RMSE = MAE.

Since we have RMSE and MAE value calculated, we can understand how well the deep learning frameworks performed in forecasting the books' rankings that we expect. Table III below shows the comparison results for the 5 algorithms after we subtract the MAE with RMSE. Among all, GAN has the least difference from both test and predict dataset. It shows that the variance in GAN individual error is the smallest among all the neural networks. Out of surprise, the predict set number is 30% lower as compared to the test set. This shows that GAN predictions on the books' rankings versus the actual rankings are even closer to each other.

The tables below show the cross-validated training results for all the models. In our research, the training and testing datasets are randomly split into five different sets for modeling. MAE and RMSE results are shown in Tables IV and V, respectively after 5 rounds of cross-validations. Table VI shows the values of RMSE–MAE on all 5 algorithms. This time around, in the cross-validation training, GAN performed the best in all areas. On average, GAN scores the lowest in both MAE and RMSE results.

TABLE. III. RMSE SUBTRACT MSE SCORES

Neural Network	Test RMSE - MAE	Predict RMSE - MAE
MLP	84272	56600
DBN	55228	91936
1DCNN	56902	60349
2DCNN	118475	79628
GAN	**50887**	**35789**

TABLE. IV. CROSS VALIDATED MAE RESULTS

MAE Results	*k* = 1	*k* = 2	*k* = 3	*k* = 4	*k* = 5	Average
MLP	731664	778127	918803	623871	827868	776067
DBN	105952	136709	123384	102635	112538	116243
1DCNN	754233	794243	930235	689700	644626	762608
2DCNN	754233	794243	930234	689700	644626	762607
GAN	85278	107280	88790	75405	82809	87912

TABLE. V. CROSS VALIDATED RMSE RESULTS

RMSE Results	*k* = 1	*k* = 2	*k* = 3	*k* = 4	*k* = 5	Average
MLP	2.04e6	1.83e6	2.16e6	1.43e6	1.82e6	1.86e6
DBN	2.73e5	4.37e5	3.08e5	1.78e5	2.66e5	2.92e5
1DCNN	1.82e6	1.97e6	2.19e6	1.59e6	1.51e6	1.81e6
2DCNN	1.82e6	1.97e6	2.19e6	1.59e6	1.51e6	1.81e6
GAN	1.37e5	1.51e5	1.52e5	1.24e5	1.26e5	1.38e5

TABLE. VI. RMSE SUBTRACT MSE (CROSS-VALIDATED)

RMSE - MAE	*k* = 1	*k* = 2	*k* = 3	*k* = 4	*k* = 5	Average
MLP	1307696	1054275	1241158	810764	988374	1080453
DBN	167086	300770	184280	74945	153155	176047
1D CNN	1061617	1173695	1261903	899627	863283	1052025
2D CNN	1061617	1173695	1261904	899627	863283	1052025
GAN	51941	44020	63187	48897	43551	50319

VIII. CONCLUSION

Generative Adversarial Network, GAN has gained a lot of attention in researches and a lot of momentums in publications. Ever since it was introduced in the year 2014, the numbers of publishing GAN related papers and journals are increasing over the past few years. In addition, many sub categorical types of GAN have been invented for example Deep Convolutional Generative Adversarial Networks (DCGAN), Wasserstein Generative Adversarial Networks (WGAN) and Conditional Generative Adversarial Networks (CGAN) that helps the GAN training to be more stable and easier [20]. However, the majority of them are focused on multidimensional unsupervised learning, especially in computer vision processing. Until recently, CGAN starts to capture attention in regression and time series prediction. Our research also shows that CGAN is capable of performing the regression predictions integrated with time series information effectively, for example, the books' rankings prediction that contained the books' features and their past rankings.

Initially, from the original books' ranking dataset, we have more than 100,000 books' rankings being captured over the 77 weeks. Secondly, in the original books' features dataset, we have collected nearly 50,000 individual rows of books'

attributes. In the best-case scenario, we should have almost 50,000 different books that consist of their own attributes and rankings that collected over one year. After we performed the necessary cleaning and combined all the dataset into one single meaningful dataset base on similarity features, we are left with about 1900 books. As we further split the dataset for training, testing and predicting, the training set is left with no more than 1100 books. This shows that a lot of the book titles are not being captured in the feature set that ends up lead to very few datasets for training our algorithms.

The other challenging part that we faced is the books' rankings range. The gap within the books' rankings is too wide. The best and worst performance book ranking can vary from top 1 to bottom 15 million in rank values. To make things even worse, based on the boxplot, the books' rankings are not evenly spread as well. The other finding we noticed while performing the Pearson correlation coefficient matrix is the book's attributes do not have a strong relationship with the final ranking values. Many of the correlation coefficient, r fall below ±0.1 and near to zero. According to The Basic Practice of Statistics, for any r value fall below ±0.3 indicates a weak relation between the two attributes [26]. The best positive and negatively correlated values are not more than 0.23 and -0.15. These indicate that the books' features collected have a weak relationship with the books' rankings. Even if we include the weak attributes in building the deep learning frameworks, they do not bring significant weights in predicting the final ranking values.

With only 1100 rows of books that possessed weak features, and having a very big spread of rankings values that appeared between the 1100 books, it is not the models to blame on underperforming. These also lead to huge numbers with many digits are appearing in the MAE and RMSE scores on all the deep learning frameworks in our research. With just a few bad predictions and cause a huge difference between the predicted ranking and expected values, the calculated loss functions will spike upward significantly.

In order to strengthen and benefit the model training, the first approach is to increase the number of datasets available for training. It is very important as the books' rankings have a large variation. Bringing up the numbers in the training dataset will help to narrow the gap between one book with another. With more datasets, the books' features and the rankings can become more distinctive as well. Secondly, it is to collect and identify more books' features that possess either a strong positive or a strong negative correlation with the ranking. Currently, the book's features are weak and the prediction of the book's final ranking figure relies heavily on the past rankings collected over a year. By inserting a strong correlation coefficient book's attributes, hopefully, the book's attributes and historical rankings can achieve a better balance in the weight of the mathematical functions in the deep learning framework.

For future research, we would suggest continuing to explore the supervised and semi-supervised learning with the GAN framework. There are many other areas that we can continue to study and leverage on the GAN to build a linear, nonlinear, or logistic regression type of mathematical algorithms. To have a better understanding of different GAN behaviours in regression predictions, moving forward, we probably can explore by including more types of GAN frameworks in our studies and compare the performance among themselves.

For both the generator and discriminator networks that appear in the GAN framework, both the neural networks are working and competing at the same time as the zero-sum game kind of relationship. Due to this, GAN typically requires an extensive amount of training, large computing power and long hour of processing in order to achieve the desired results. Henceforth, there is still a big room of improvement available for GAN to move forward and evolved to the next level with better capabilities and higher efficiencies.

REFERENCES

[1] Flood, A., 2011. How self-publishing came of age. The Guardian.

[2] Cadavid, J.P.U., Lamouri, S., Grabot, B., 2018. Trends in Machine Learning Applied to Demand & Sales Forecasting: A Review 11.

[3] Chen, P.-Y., Wu, S., Yoon, J., 2004. The Impact of Online Recommendations and Consumer Feedback on Sales 15.

[4] Chevalier, J.A., Mayzlin, D., 2006. The Effect of Word of Mouth on Sales: Online Book Reviews. J. Mark. Res. 43, 345–354.

[5] Saha, S., 2018. A Comprehensive Guide to Convolutional Neural Specht, D.F., 1991. A general regression neural network. IEEE Trans. Neural Netw. 2, 568–576. https://doi.org/10.1109/72.97934.

[6] Jeon, Y., Cho, C., Seo, J., Kwon, K., Park, H., Chung, I.-J., 2017. Rule-Based Topic Trend Analysis by Using Data Mining Techniques, in: Park, J.J., Chen, S.-C., Raymond Choo, K.-K. (Eds.), Advanced Multimedia and Ubiquitous Engineering. Springer Singapore, Singapore, pp. 466–473. https://doi.org/10.1007/978-981-10-5041-1_75.

[7] Dou, H., Zhao, W.X., Zhao, Y., Dong, D., Wen, J.-R., Chang, E.Y., 2018. Predicting the Popularity of Online Content with Knowledge-enhanced Neural Networks 8.

[8] Wang, X., He, X., Wang, M., Feng, F., Chua, T.-S., 2019. Neural Graph Collaborative Filtering. ArXiv190508108 Cs. https://doi.org/10.1145/3331184.3331267.

[9] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets 9.

[10] Aggarwal, K., Kirchmeyer, M., Yadav, P., Keerthi, S.S., Gallinari, P., 2019. Conditional Generative Adversarial Networks for Regression. ArXiv190512868 Cs Stat. (10).

[11] Zhang, K., Zhong, G., Dong, J., Wang, S., Wang, Y., 2019. Stock Market Prediction Based on Generative Adversarial Network. Procedia Comput. Sci. 147, 400–406. https://doi.org/10.1016/j.procs.2019.01.256.

[12] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., 2000. Step-by-step data mining guide. SPSS Inc 76.

[13] Harwell, M.R., Gatti, G.G., 2001. Rescaling Ordinal Data to Interval Data in Educational Research. Rev. Educ. Res. 71, 105–131. https://doi.org/10.3102/00346543071001105.

[14] Xu, Y., Goodacre, R., 2018. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. J. Anal. Test. 2, 249–262. https://doi.org/10.1007/s41664-018-0068-2.

[15] Yeager, K., 2020. LibGuides: SPSS Tutorials: Pearson Correlation [WWW Document]. URL https://libguides.library.kent.edu/SPSS/PearsonCorr (accessed 1.8.20).

[16] Nyce, C., 2007. Predictive Analytics White Paper. Am. Inst. CPCU 24.

[17] Patterson, J., Gibson, A., 2018. Getting started with deep learning [Book]. O'Reilly Media, Inc.

[18] Mohamed, A., Dahl, G., Hinton, G., 2009. Deep Belief Networks for phone recognition 9.

[19] Bouvrie, J., 2006. Notes on Convolutional Neural Networks 8.

[20] Neff, T., 2018. Data Augmentation in Deep Learning using Generative Adversarial Networks 113.

[21] Wu, Y., Yang, F., Liu, Y., Zha, X., Yuan, S., 2018. A Comparison of 1-D and 2-D Deep Convolutional Neural Networks in ECG Classification 4.

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting 30.

[23] Oliehoek, F.A., Savani, R., Gallego-Posada, J., van der Pol, E., de Jong, E.D., Gross, R., 2017. GANGs: Generative Adversarial Network Games. ArXiv171200679 Cs Stat.

[24] Mirza, M., Osindero, S., 2014. Conditional Generative Adversarial Nets 7.

[25] Chai, T., Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geosci. Model Dev. 7, 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014.

[26] Mindrila, D., Balentyne, P., Ed, M., 2012. Scatterplots and Correlation. Basic Pract. Stat. 6th Ed, Chapter 4 14.