Attention-Based Neural Network: A Novel Approach for Predicting the Popularity of Online Content

Conference Paper · May 2019

DOI: 10.1109/HPCC/SmartCity/DSS.2019.00058

5 authors, including:

Minh-Tri Nguyen
Aalto University
9 PUBLICATIONS **18** CITATIONS
SEE PROFILE

Duong H Le
Ho Chi Minh City University of Technology (HCMUT)
4 PUBLICATIONS **9** CITATIONS
SEE PROFILE

Masato Yoshimi
Doshisha University
82 PUBLICATIONS **380** CITATIONS
SEE PROFILE

Nam Thoai
Ho Chi Minh City University of Technology (HCMUT)
109 PUBLICATIONS **621** CITATIONS
SEE PROFILE

Attention-based Neural Network: A Novel Approach for Predicting the Popularity of Online Content

Minh-Tri Nguyen*, Duong H.Le*, Takuma Nakajima†, Masato Yoshimi‡ and Nam Thoai*
*Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology
Vietnam National University Ho Chi Minh City, Vietnam
Email: {nmtribk, 1610580, namthoai}@hcmut.edu.vn
†Service Strategy Sector and ‡Strategic Technology Center, TIS Inc., Shinjuku-ku, Tokyo, Japan
Email: {nakajima.takuma, yoshimi.masato}@tis.co.jp

***Abstract*—Since the rate at which new content is uploaded to the Internet has reached unprecedented marks, knowing the popularity of online content, especially video, is of importance for network management, recommendation schemes, service design, advertising planning, and so on. Despite the fact that various models have been developed, few of them address the short-term popularity prediction. Toward this goal, we exploit the self-attention mechanism of the Transformer, a state-of-the-art model in neural machine translation, to forecast the values of multiple time series in the near future. Specifically, we propose an attention-based non-recursive neural network, a novel model that entirely dispenses with recurrence and convolutions, for time series prediction. Since our model is the combination of the input attention mechanism in the dual-stage attention-based recurrent neural network (DA-RNN) and the self-attention of Transformer, it is able to adaptively select the most relevant input sequences as well as capture the long-term dependencies across previous time steps to make the prediction. The experiments show the root mean square errors (RMSEs) achieved by our model are only 6.06 and 3.60 when testing on the NASDAQ 100 dataset and the views count of the top most popular videos on Youtube respectively, while the RMSEs of DA-RNN are 8.52 and 12.31. Hence, our model outperforms the baseline not only in time series prediction but also in contents popularity prediction aspect.**

***Index Terms*—Transformer, deep learning, time series prediction, Youtube video.**

1. Introduction

With the emergence of social network and the explosion of internet usages, an enormous and ever growing amount of online content has been brought into the digital world. In this context, video contents are determined to be a dominant cause of network congestion as it would account for more than 80% of total Internet traffic by 2020 [1]. However, it has been proved that user attention is allocated in a skewed fashion since most contents get a few views and downloads, whereas a few others receive massive user attention [2]. Knowing the popularity of each content in the future, the service providers can exceedingly benefit from designing appropriate strategies and recommendation schemes assisting their customers in reaching the most relevant and popular contents. From the internet service aspect, the network operators can proactively manage the distribution and the cache replacement policies for online contents within their infrastructures. Hence, predicting the popularity of online contents, especially videos, is of great importance to support and drive the design and management of various services. While it is possible to predict the long-term popularity of online videos [2], [3], [4], it has been hard to predict an amount of user attention to given ones in the near future or, in short, make the short-term prediction. In this paper, we propose a novel time series model to address this problem.

In general, time series prediction is the use of a model to predict future values based on previously observed values. In other words, the predicting involves taking models to fit on historical data and applying them to forecast the future values of the series. Time series prediction is a classic problem in many domains, with wide-ranging and high-impact applications. There are many different methods for performing time series prediction, for example, exponential moving average [5], [6], autoregressive model [7], polynomial regression [8], autoregressive integrated moving average [9], [10] and so on. In the past few years, Recurrent Neural Network (RNN) and its variances have outperformed the traditional methods. Thus, they have played the central role and become the standard for many time series analysis problems.

Recently, several efforts on Natural Language Processing (NLP) addressing some problems such as sentiment analysis, sequence-to-sequence translation, result in many superior mechanisms and techniques that can be widely applied in many other areas. Notably, the attention mechanism in combination with Long-Short-Term Memory (LSTM) recurrent network, which was proposed in a recent study [11], has reached a new state-of-the-art in time series prediction.

Inspired by the ideas of applying attention mechanism of neural machine translation into time series prediction, we propose a Transformer based model called attention-based non-recursive neural network (ANRNN) to address the short-term prediction. Then, we evaluate our model per-

formance on two real datasets which are NASDAQ 100 and the views count of the top most popular videos on Youtube. Our approach shows a significant reduction in prediction errors compared to two baselines. Since our model does not require any recurrent or convolutional element to make predictions on multiple sequences simultaneously, it is much more parallelizable than other existing methods, which leads to the significant reduction in training/testing time.

The rest of the paper is organized as follows. Section 2 outlines some efforts that have been done in the field of time series prediction and long-term popularity prediction. Section 3 presents our model with some background knowledge related to its design. Section 4 discusses the empirical results of our model on the two real datasets compared to the baselines. Finally, Section 5 draws the conclusions.

2. Related Work

2.1. Long-term popularity prediction

The field of predicting online contents popularity was pioneered by the initiative of Szabo and Huberman [2]. The paper shows the proofs of a strong linear correlation between the long-term popularity and the early popularity on the logarithmic scale. Based on the property, the authors proposed a simple log-linear model to predict the popularity of given online content in the future. The model feasibility was tested on various datasets including Youtube videos [12], Digg stories [2] and so forth. Following this idea, Pinto et al. [3] proposed two multivariate regression models that make predictions using the daily sample of contents popularity measured up to the given reference date. The experiment results proved that their models achieved a reasonable accuracy on Youtube dataset.

Recently, Li et al. [4] introduced a novel model that can capture the popularity dynamics based on early popularity evolution pattern and future popularity burst prediction. In this work, the authors not only use some basic early popularity measurements but also consider the characteristic of individual video as well as its popularity evolution pattern as the input of their model. In addition to the regression-based methods, some other techniques such as reservoir computing [13], time series analysis [14] are also applied to improve the performances.

Despite achieving initial results, the aforementioned studies mainly focused on predicting the long-term popularity of the given content. To address the problem in both long-term and short-term, a recent work [15] proposed a simple artificial neural network (ANN) to predict the popularity of scientific datasets at the Large Hadron Collider at CERN. Since the model is not robust enough to accurately predict the popularity of an item, it is mainly used for classification.

As accurately predicting the popularity of online contents in the near future is not a trivial task. Our work here is exploiting state-of-the-art techniques in time series prediction to address the problem.

2.2. Time series prediction

Time series prediction algorithms have been extensively researched in recent years and applied to solve many critical problems in various areas such as financial market prediction [16], weather forecasting [17], predicting the next frame of a given video [18] and complex dynamical system analysis [19]. Although these works have shown their effectiveness for various real-world applications, most of these models are parameterized by a Recurrent Neural Network - LSTM or its variants.

In these models, recurrent neural networks (RNNs) compute a hidden state h_t, as a function of their input at time t and a previous hidden state h_{t-1}, capturing relative and absolute positions along the time dimension directly through their sequential structure. In theory, RNNs are generally capable of handling long-term dependencies, however, in practice vanilla recurrent neural network suffered from the gradient vanishing problem and cannot learn the dependencies in long sequences [20]. Long Short Term Memory network, also known as LSTM, is a special kind of RNN, capable of learning long-term dependencies. Initially introduced in Hochreiter's study [21], it works very well on a large variety of problems.

Fundamentally, to correctly predict the value at a time step, we sometimes have to base on the values at several time steps that are far away from the target time step. This is identical to the long-range dependencies in Language Modelling. To solve this problem, several efforts have been made to exploit the potential of LSTM in time series prediction [11]. However, a recent study [22] empirically shows that LSTM cannot handle very long dependencies in language models since it is only capable of using about 200 tokens of context on average. Moreover, the network is only high-sensitive to the order of the most recent words, which implies that the LSTM cannot capture the long-range dependencies very well. This fact opens room for further improvement.

Basically, RNNs sequentially process each input at each time step, after processing the entire input, the final hidden vector having fixed size will be used as a context vector to predict the value in the next time step. This fixed size context vector appears to be the bottleneck for these types of models. While LSTM makes it challenging for the model to deal with long-term dependencies, attention mechanism allows the model to focus on the relevant parts of the input sequence as needed and achieve better results on long input sequences. Attention mechanism, a method used in modern deep learning models, was proposed by Bahdanau et al. [23] to focus on the long-term dependencies in many NLP tasks. Later on, many sufficient approaches for attention based method were also provided by Luong et al. [24].

Though working well, the LSTM and the conventional attention mechanisms may not be scalable for applications manipulating a large amount of data as they have to compute everything sequentially. Some later studies have been done to address this issue by utilizing the convolutional neural network (CNN) to reduce the computation cost [25], [26], [27]. In contrast with traditional methods, our approach

entirely dispenses with recurrent and convolutional neural networks. Since our model is built based on the architecture of Transformer [28], it results in the improvement in both accuracy and training/testing time.

3. Model Architecture

3.1. Background

3.1.1. Sequence-to-sequence. The sequence-to-sequence model was introduced in the field of neural machine translation [29]. Fundamentally, it aims to transform a source sequence into the target sequence where both sequences can be of arbitrary lengths. In this work, we propose a novel model to transform sequences containing historical values of input series into sequences containing values of these series in the next time step.

The sequence-to-sequence model often contains 2 submodules: encoder and decoder where the encoder will encode the input into a fixed size context vector which will be used by the decoder to generate the output. Although it worked very well in I.Sutskever's study [29], the quality of the model dramatically decreases as processing the longer sequence since it may forget the previous parts when it finishes processing on the whole input. Thus, the attention mechanism was proposed to address the problem [23].

3.1.2. Attention mechanism. Instead of sequentially encoding the whole input sequences into fixed size vectors, also known as hidden states, and using the last hidden state as the context vector, the attention mechanism creates shortcuts between the context vector and the entire input sequence. The weights of these shortcut connections are customizable for each output element. Over time, several variants of attention mechanism have been proposed to solve some specific problems. These mechanisms include content-based attention [30], addictive (also called "concat") attention [23], location attention [24], scale-dot product attention [28]. Below is a brief summary of several popular attention mechanisms and corresponding alignment score functions.
Content-based attention:

$$\text{score}(\mathbf{s}_t, \mathbf{h}_t) = \text{cosine}[\mathbf{s}_t, \mathbf{h}_t]$$

Dot product attention:

$$\text{score}(\mathbf{s}_t, \mathbf{h}_t) = \mathbf{s}_t^T \mathbf{h}_t$$

Addictive product attention:

$$\text{score}(\mathbf{s}_t, \mathbf{h}_t) = \mathbf{v}_\alpha^T \tanh(W_\alpha[\mathbf{s}_t; \mathbf{h}_t])$$

where h_t, s_t are the hidden state and the cell state of the RNN model at time step t. Then, W_α,$\mathbf{v}_\alpha$ are the learned parameters.

Self-attention (also called intra-attention) [31] is a mechanism that performs shallow reasoning with memory and attention. In contrast to inter-attention, self-attention requires the model to compute the attention score of the different positions within a single sequence. Recently, it has been successfully applied in a variety of tasks including language modeling [31], sentiment analysis [31], natural language inference [31], [32], abstractive summarization [33], learning task-independent sentence representations [34] and so on.

3.1.3. The Transformer. One of the limitations of models belonging to the RNN family is that they sequentially compute each time step, which leads to long training and inference time. Much effort has been made to address this issue like replacing the RNNs with very deep CNNs to capture the long-term dependencies. For instance, Gehring et al. [27] investigated convolutional layer for sequence-to-sequence tasks, Zeng et al. [25] exploited a convolutional deep neural network to extract lexical and sentence level features, Conneau et al. [26] applied very deep convolutional nets to text processing.

Unlike those approaches, in the recent study, Vaswani et al. [28] proposed a novel model called Transformer. As common sequence-to-sequence models, the Transformer's architecture is built based on encoder-decoder structure. However, its encoder is composed of N stacked identical layers. Each layer has two sub-layers which are the self-attention layer and the fully connected feed-forward layer. In the same manner, the decoder consists of M stacked identical layers, but its elemental layer has 3 sub-layers. In addition to the two sub-layers in the encoder, the decoder has an encoder-decoder attention layer to perform attention on source sequence representation. As entirely eliminating recurrent and convolutional connections as well as applying the self-attention mechanism to capture the long-term dependencies, the Transformer has been proved to reach new state-of-the-art in translation quality.

3.2. ANRNN Model

The recent study [11] has succeeded in applying LSTM with attention to time series prediction. It presents a great opportunity as well as a challenge in utilizing the state-of-the-art techniques of NLP to address some problems in time series prediction. Aiming at increasing the accuracy as well as the efficiency, we propose an attention-based non-recursive neural network (ANRNN), a novel model for time series prediction. Instead of predicting single value, our model can predict n values of n driving series at once. Its structure is described in Fig. 1.

Like most of competitive neural sequence transduction models [28], ANRNN model has an encoder-decoder structure. However, its encoder here is integrated with an input attention layer to highlight the relevant driving series before they are fed into the core network. The encoder now maps an input $E = (e^1, e^2, ..., e^{T-1})$ to an encoded values $Z = (z^1, z^2, ..., z^{T-1})$, where e^t, z^t is a vector containing respectively the output of input attention layer and the encoded values of n driving series at time step t; $e^t, z^t \in \mathbb{R}^n$, and T is the size of the sliding window. Given Z, the decoder then generates a vector output $y^T = (y_1, y_2, ..., y_n)$ which contains predicted values of n driving series at the last time step of the sliding window.

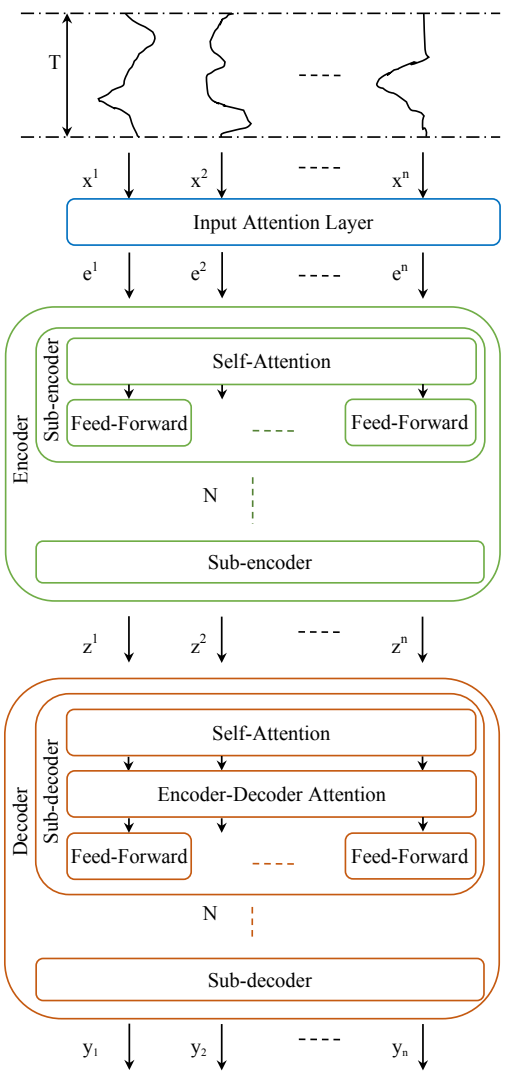


Figure 1: The graphical illustration of Attention-based Non-Recursive Neural Network model architecture.

Similar to the Transformer, our encoder is the stack of N identical layers. Each layer consists of two sub-layers named self-attention and feed-forward layer. At self-attention layer, attention is obtained by the following formula [28]:

$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d_k}}V) \quad (1)$$

where $Q = EW^Q; K = EW^K; V = EW^V$, $Q \in \mathbb{R}^{(T-1)\times d_k}, K \in \mathbb{R}^{(T-1)\times d_k}, V \in \mathbb{R}^{(T-1)\times d_v}$ are queries, keys and values matrices. Then, $W^Q \in \mathbb{R}^{n\times d_k}, W^K \in \mathbb{R}^{n\times d_k}, W^V \in \mathbb{R}^{n\times d_k}$ are weight matrices that the model has to learn. In this case, we also set $d_k = d_v = 64$ [28]. The self-attention is applied to this model as it allows the model to look at other time steps in the input sequences for clues to make a better encoding for the current processing time step. In contrast to the recurrent neural network, self-attention does not contain any recursive step, which makes it easy to be parallelized that significantly lower the training time. After the self-attention layer is a feed-forward network which consists of a fully connected ANN. It has two linear transformation layers and a ReLU activation in between. The feed-forward layer is applied to each time step identically but separately as it uses different parameters for each time step.

Besides, all sub-layers are followed by a layer normalization [35] which is commonly used to normalize the activities of the neurons as a method to decrease the training time. Specifically, the layer normalization computes the mean and variance from all of the summed inputs to the neurons in a layer on a single training case to normalize the neurons. Moreover, as the model is the stacked of many layers, the training become hard because of the gradient vanishing problem. Therefore, residual connections [36] are employed around each of two sub-layers to allow gradients to pass through a network directly without passing through non-linear activation functions. In the same manner, the decoder is a stack of N identical layers, but each layer has an additional encoder-decoder attention layer between two others to perform attention on source sequence representation.

As our ANRNN model is the combination of two attention mechanisms, it can pay attention to not only the most relevant driving series but also the critical time steps. To further understand the model, we conduct some experiments on two real datasets, which will be discussed in the next session.

4. Experiments

In this section, we discuss our experimental results and compare the predictability of our ANRNN model against two other methods. The first one is a conventional fully connected ANN (FC-ANN), which was mentioned in Hushchyn's study [15] as it could predict the popularity of dataset used at the Large Hadron Collider at CERN. In our work, we use an ANN which contains several hidden layers as described in the following formulas:

$$h_{2k} = ReLU(h_{2k-1}) = ReLU(W_{2k-1}i^{2k-1} + b_{2k-1}) \quad (2)$$

$$o = W_n i_n + b_n \quad (3)$$

where n is the number of layers, $n = 2k + 1$ and $k = 0, 1, ..., n/2$. i^k is the input at layer h_k (k^{th} layer), o is an output of the model, W_k and b_k are parameters to learn at layer h_k. By repeating the experiment with various values of n and the size of each hidden layer in range {16, 32, 64, 128, 256, 512}, we found that $n = 5$, size of h_1, h_5 equal to 64 and size of h_3 equal to 128 gives the best results.

The second baseline is DA-RNN, proposed in the recent study [11]. In this study, the superiority of the DA-RNN model was demonstrated in comparison to four baseline methods including the autoregressive integrated moving average (ARIMA) [37], the non-linear autoregressive exogenous recurrent neural network (NARX RNN) [38],

the encoder-decoder network [39] and the attention-based encoder-decoder network (attention RNN) [23].

For each experiment, we repeat the same process multiple times and report the average results.

4.1. Dataset

The first dataset is NASDAQ 100 which was used in the work of Quin et al. [11] as it is commonly used for time series prediction and analysis. The dataset consists of the stock prices of 81 major corporations under NASDAQ 100. The data was collected minute-by-minute from July 26, 2016, to December 22, 2016 (105 days in total). Each day contains about 390 data points from the opening to the closing of the market. We split this dataset into small chunks consisting of 700-800 data points as we mainly focus on short-term prediction.

The second dataset was collected from Youtube, one of the largest video hosting service, via its application programming interface [12]. Up to 2018, Youtube has more than 30 million active users and about 5 billion videos watched on a daily basis and it is also ranked as the second-most popular site in the world, according to Alexa Internet [40]. In this work, we collect the data in several 10-day periods, February 21 to March 2, 2019; March 2 to March 12, 2019; and March 12 to March 22, 2019. Due to the limited policy of Youtube, each data set contains hourly records of only 50 videos/country. These videos are the most popular contents at the beginning of collecting process within 20 countries including the US, Japan, Russia, China, India and so on, where most of the site's visitors are located [40]. Hence, our dataset contains the views count of more than 2.500 different videos. We estimated that the average number of views of each country is about 430 million views per hour. Since the views count of some videos does not appear to be updated more often than once an hour, we conducted our experiment on some subsets that have views count regularly updated.

To evaluate the effectiveness of the three methods while applying in these two datasets, we apportion the data into train, test sets with an 80:20 split and consider two scale-dependent evaluation metrics which are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Specifically, we denote y_t is the true value and $\hat{y}_t$ is the predicted value at time t. Then, RMSE and MAE are calculated by the following formulas: $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_t^i - \hat{y}_t^i)^2}$; $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_t^i - \hat{y}_t^i|$. Since the wide range of value in the dataset would make it difficult for the model to converge, as well as evaluating the performance, all data are normalized.

4.2. Parameter Sensitivity

In the first experiment, we investigate the effect of ANRNN's parameters on the predicting result by keeping track of the RMSE values during the learning process. As choosing the number of time steps ws (the window size) is greatly dependent on the characteristic of the dataset, we conducted a grid search over $ws = \{2, 3, ..., 20\}$ and found that $ws = 9$ and $ws = 3$ give the best performance over the validation on test set of NASDAQ 100 and Youtube dataset respectively. Then, the model is trained with $N \in \{1, 2, ..., 10\}$, and $hs \in \{32, 64, 128, 256, 512\}$, where N is the number of encoder-decoder layers, and hs is the size of the hidden layers in feed-forward and self-attention block.

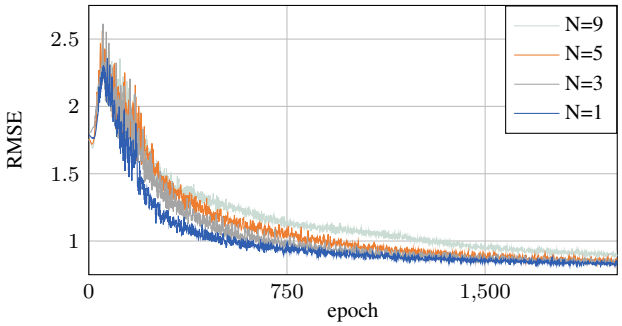


Figure 2: RMSE comparison among different number of stacked layers in training with Youtube dataset.

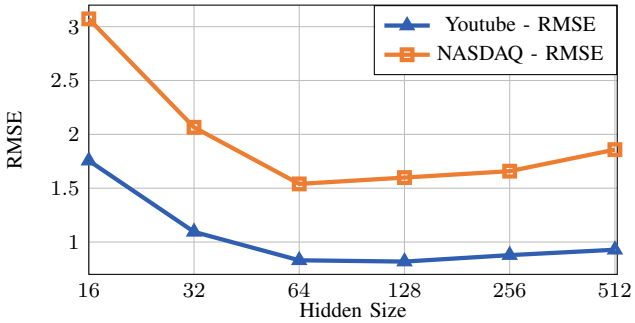


Figure 3: RMSE vs. hidden size of input attention layer and feed-forward nets of each sefl-attention layer in training with Youtube and NASDAQ 100 dataset.

It is apparent from Fig. 2 that the model is converged quickly after about 500 epochs when $N = 1$ and the convergence rate decreases as N increases. In fact, increasing the number of layers means increasing the complexity of the model that would significantly inflate the computation cost. In this case, the model does not benefit from increasing the complexity and it can be seen that the model is not converged after 2000 epochs when N is greater than 9.

After that, by fixing $N = 1$, we plot the RMSE versus various hidden sizes (hs) when the model converged. The

TABLE 1: Mean correlation among series in 2 datasets.

Method	**Correlation Coefficient**	
	NASDAQ 100 dataset	Youtube dataset
Pearson [41]	0.07785	0.47919
Spearman [42]	0.08259	0.53298
Kendall [43]	0.07437	0.40476

TABLE 2: RMSE loss comparison between DA-RNN and ANRNN when practicing with NASDAQ 100 dataset.

Algorithms	Train set		Test set	
	RMSE	MAE	RMSE	MAE
DA-RNN	3.45776	2.38940	8.52048	7.06531
ANRNN	1.53092	1.16549	6.05537	4.97235

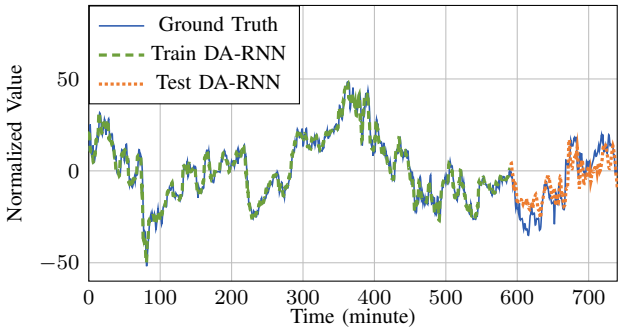
hidden size does not appear to bring any remarkable changes when it is greater than 64. Repeating the experiment multiple times, we observe that the model with the hidden size of 128 always achieves the RMSE that is a bit lower than the one with the hidden size of 64 while training on the Youtube dataset, the opposite behavior is observed on the NASDAQ 100 dataset. The average RMSEs over those experiments are shown in Fig. 3.

In short, our model only needs one encoder-decoder layer to make predictions on both datasets and it would be worse as the number of layer increases. Although setting the hidden size is not so sensitive to the dataset, it needs to be tested multiple times to achieve better results. In some case, we may consider the tradeoff between the accuracy and the hidden size of the model.

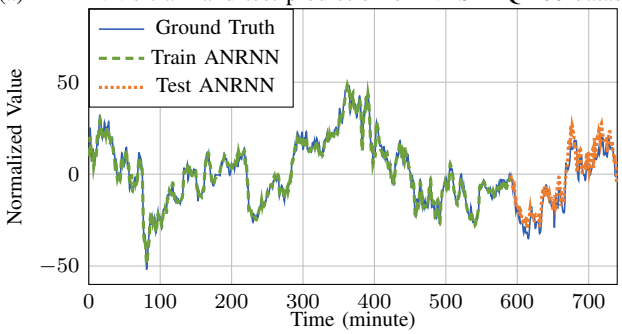
4.3. Time series prediction

In perspective of time series prediction, we compare our model with the DA-RNN when practicing with NASDAQ 100 dataset to investigate how well the models follow the changes in single driving series. Since most of the time series in this dataset are not cyclical and the relations between them are complicated, a simple neural net like FC-ANN is unable to make a good prediction. As reported in Table. 1, the correlation coefficients among series in NASDAQ 100 dataset is small, about 0.07 to 0.09, which means they do not have any linear relation. Thus, in Fig. 4, we only plot the predicted results of the two other models on both train and test set. The DA-RNN model was proved that it generally fits the ground truth much better than the four mentioned baselines when practicing on the whole dataset [11]. To re-evaluate this result, we did conduct the experiment using the verified public source code [44] and it is illustrated in Fig. 4 (a), where we plot its predictions on about 750 data points of a single time series. Similarly, the prediction conducted by our model is plotted in Fig. 4 (b). Since both models fit very well on the train set, we consider the RMSE and MAE values to evaluate their performance. As seen in Table. 2, our model outperforms DA-RNN with the lower RMSE and MAE values in the train set.

Since both models use the input attention mechanism to adaptively select the most relevant input time series, the only difference between them is the second attention mechanism used to capture the long-term dependencies across previous time steps. Specifically, DA-RNN applies LSTM-based recurrent neural network to encode as well as decode input information. This module generates a sequence of hidden states $h = \{h_1, h_2, ..., h_n\}$, in which, h_t stores input information at time step t, and uses them to compute the next hidden state, enabling the model to adaptively select the



(a) DA-RNN's train and test prediction on NASDAQ 100 dataset.



(b) ANRNN's train and test prediction on NASDAQ 100 dataset.

Figure 4: NASDAQ 100 Index (normalized values) vs. time and predicting values of DA-RNN as well as ANRNN on NASDAQ 100 dataset.

most relevant hidden state across all the time steps as well as accumulate information to make a prediction. Meanwhile, our model uses the self-attention mechanism which has been proved to reach the state-of-the-art in sequence-to-sequence translation [28]. This mechanism gives our model an ability to pay more attention to the most critical dependencies among all time steps. By exporting the attention weight while running our model, we observe the noticeable difference in attention weights among all time steps. It means that each previous time step has a different contribution to computing future values. As learning those attention weights better, our model shows its superiority in performance compared to DA-RNN. This fact is further illustrated in the test set, where the RMSE of our model and DA-RNN are about 6.06 and 8.52 respectively.

4.4. Predicting content popularity

To demonstrate the effectiveness of our model in predicting content popularity, in this practice, we compare the predictability of the three models on Youtube dataset. Since some videos in the same category as well as having the same owner often have the same popularity evolution pattern, their popularity logs often have linear relations. This fact is also shown in Table 1, where the mean correlation coefficient is about 0.40 to 0.53. Moreover, as reported in Szabo's study [2], the popularity of most of Youtube's contents have daily

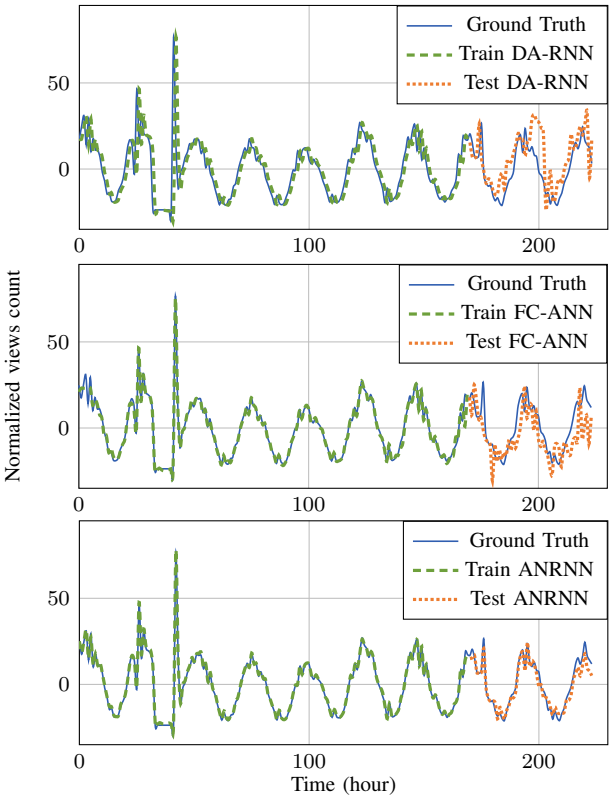


Figure 5: Video's views count vs. time and predicting values of DA-RNN, FC-ANN and ANRNN on Youtube dataset.

and weekly cycles, since the users are more active in the day time and less active in the weekend. It makes prediction become easier for some simple model like FC-ANN.

Considering Fig. 5, we can observe that FC-ANN and ANRNN generally fit the ground truth in the train set, while the outcome of DA-RNN appears to be a delayed sequence of the ground truth. Here, DARNN's result has been biased by the value of the last time step, which proves that its attention mechanism works ineffectively in this case.

For a better understanding of the model performance, we also report the RMSE and MAE values produced by each model in Table 3. Although the FC-ANN gives an impressive result on the train set with RMSE value much lower than DA-RNN, its outcome in the test set is not really satisfied with RMSE about 11.5. Notably, the DA-RNN conducts the worst result among the three models. The explanation for this is that the RNN sequence-to-sequence models have not been able to attain a remarkable performance in small-data regimes [45]. In contrast, our model appears to be very effective as it achieves the lowest RMSE and MAE in both train and test set.

In summary, the above empirical results have shown that ANRNN outperforms two baseline methods when practicing with two real datasets. As our model relies entirely on input attention and self-attention mechanism to map its input and output without using RNN or CNN, it is more parallelizable and requires less training and testing time.

TABLE 3: RMSE loss comparison between DA-RNN, FC-ANN and ANRNN when practicing with Youtube dataset.

Algorithms	**Train set**		**Test set**	
	RMSE	MAE	RMSE	MAE
DA-RNN	12.18310	6.46745	12.31086	9.69878
FC-ANN	2.31416	1.28454	11.51251	8.96781
ANRNN	0.81684	0.58971	3.59979	2.80669

5. Conclusion

In this work, we present a novel model, ANRNN, which is the combination of two state-of-the-art attention mechanisms, to address short-term prediction, especially in predicting online contents popularity. The input attention enables our model to effectively select the relevant input series while the self-attention of Transformer can support the model to capture the important long-range dependencies across all time steps. The experiment results have demonstrated that our model can outperform state-of-the-art methods for time series prediction when being evaluated in two real datasets which are NASDAQ 100 and Youtube dataset. Since our model is highly capable of being parallelized as well as predicting multiple sequences simultaneously, not only the prediction accuracy increases but also the training time is dramatically reduced. Hence, the model proposed in this work would be potential and beneficial in predicting content popularity and applying to other related fields.

Besides, our experiments only consider the views count of videos to make predictions. However, the Youtube dataset provides many other features such as category, owner, number of likes, dislikes and comments and so on. These make room for future experiments. Moreover, some new videos have attracted the attention of a large number of users in a short period of time, but there are not enough historical records to make a good prediction. It can also be considered as a drawback of our model. Hence, our ANRNN needs to be improved and evaluated further on many other datasets to demonstrate its applicability and reliability in the future.

Acknowledgments

This research was conducted within the project Optimizing Color-Based Cooperative Caching Algorithm for Telco-CDNs sponsored by TIS Inc. (IT Holding Group).

References

[1] C. V. N. I. Cisco, "The zettabyte eratrends and analysis, 2015–2020. white paper," 2016.

[2] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Available at SSRN 1295610*, 2008.

[3] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of youtube videos," in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 365–374.

[4] C. Li, J. Liu, and S. Ouyang, "Characterizing and predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 1630–1641, 2016.

[5] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[6] S. Hansun, "A new approach of moving average method in time series analysis," in *2013 Conference on New Media Studies (CoNMedia)*. IEEE, 2013, pp. 1–4.

[7] L. Harrison, W. D. Penny, and K. Friston, "Multivariate autoregressive modeling of fmri time series," *NeuroImage*, vol. 19, no. 4, pp. 1477–1491, 2003.

[8] E. Masry, "Multivariate local polynomial regression for time series: uniform strong consistency and rates," *Journal of Time Series Analysis*, vol. 17, no. 6, pp. 571–599, 1996.

[9] Y.-S. Lee and L.-I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 66–72, 2011.

[10] S. Ling and W. Li, "On fractionally integrated autoregressive moving-average time series models with conditional heteroscedasticity," *Journal of the American Statistical Association*, vol. 92, no. 439, pp. 1184–1194, 1997.

[11] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[12] "Youtube application programming interface," https://developers.google.com/youtube/. Last Accessed: March 18, 2019, 2019.

[13] T. Wu, M. Timmers, D. De Vleeschauwer, and W. Van Leekwijck, "On the use of reservoir computing in popularity prediction," in *2010 2nd International Conference on Evolving Internet*. IEEE, 2010, pp. 19–24.

[14] G. Gürsun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 16–20.

[15] M. Hushchyn, P. Charpentier, and A. Ustyuzhanin, "Disk storage management for lhcb based on data popularity estimator," in *Journal of Physics: Conference Series*, vol. 664, no. 4. IOP Publishing, 2015, p. 042026.

[16] Y. Wu, J. M. Hernández-Lobato, and Z. Ghahramani, "Dynamic covariance models for multivariate financial time series," *arXiv preprint arXiv:1305.4268*, 2013.

[17] P. Chakraborty, M. Marwah, M. Arlitt, and N. Ramakrishnan, "Fine-grained photovoltaic output prediction using a bayesian ensemble," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[18] V. Vukotić, S.-L. Pintea, C. Raymond, G. Gravier, and J. C. van Gemert, "One-step time-dependent future video frame prediction with a convolutional encoder-decoder neural network," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 140–151.

[19] Z. Liu and M. Hauskrecht, "A regularized linear dynamical system framework for multivariate time series analysis," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[20] Y. Bengio, P. Simard, P. Frasconi *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, "Sharp nearby, fuzzy far away: How neural language models use context," *arXiv preprint arXiv:1805.04623*, 2018.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[25] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao *et al.*, "Relation classification via convolutional deep neural network," 2014.

[26] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arXiv preprint arXiv:1606.01781*, 2016.

[27] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[30] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[31] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[32] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.

[33] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[34] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[35] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] D. Asteriou and S. G. Hall, "Arima models and the box–jenkins methodology," *Applied Econometrics*, vol. 2, no. 2, pp. 265–286, 2011.

[38] E. Diaconescu, "The use of narx neural networks to predict chaotic time series," *Wseas Transactions on computer research*, vol. 3, no. 3, pp. 182–191, 2008.

[39] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[40] "youtube.com traffic statistics," https://www.alexa.com/siteinfo/youtube.com. Last Accessed: March 18, 2019, 2019.

[41] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[42] L. Myers and M. J. Sirois, "Spearman correlation coefficients, differences between," *Encyclopedia of statistical sciences*, vol. 12, 2004.

[43] H. Abdi, "The kendall rank correlation coefficient," *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pp. 508–510, 2007.

[44] "A pytorch example to use rnn for financial prediction," http://chandlerzuo.github.io/blog/2017/11/darnn. Last Accessed: March 18, 2019, 2019.

[45] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in neural information processing systems*, 2015, pp. 2773–2781.