Multi-attention based Feature Embedding for Irregular Asynchronous Time Series Modelling

Swagato Barman Roy and Miaolong Yuan
Advanced Remanufacturing & Technology Centre (ARTC)
Agency for Science, Technology and Research (A*STAR)
3 Cleantech Loop, #01-01 CleanTech Two
Singapore 637143
Republic of Singapore
{barmans, myuan}@artc.a-star.edu.sg

***Abstract*—Forecasting time series values based on historic covariates has been an active area of research in statistics and machine learning. With the availability of computation resources and big data infrastructure supporting massive volume, velocity and variety, the algorithms have evolved from classic statistical learning to neural-network driven loss minimisation techniques. While state of the art attention and self-attention-transformers have shown promise of improved performance with sufficient training data, most of them fail to generalise to different problems of time-series modelling (such as classification and extremum forecasting) with asynchronously sampled covariates. This paper introduces the concept of a generalised time series embedding and transfer learning for time series (analogous to token-to-vector or image-to-vector embeddings in language and vision models respectively) that allow joint training with a unified interface. The major benefit of this work is a unified embedding model employing multi-attention for feature representation which enables benchmark performance against state of the art models from recent literature.**

I. Introduction

Predictive machine learning tasks on multi-variate time series have been cornerstones of different domains ranging from operations research to healthcare, finance, business intelligence and predictive maintenance for manufacturing operations. Most traditional research on time-series focussed exclusively on forecasting. Traditionally, statistical models like ARIMA and other auto-regressive models have been the most reliable tools for forecasting. Eventually, with the availability of big-data and massive-computing resources, neural network based computations consistently outperformed traditional models for most forecasting tasks. Among many recent innovations, the attention and transformer based models [1] continue to provide the benchmark performance for many forecasting tasks.

However, in the prevalent literature on time series forecasting, the focus has been on forecasting future values at later time-steps given historic data of multiple covariates [2]. Surprisingly little attention has been paid to related time series problems such as classification of series, prediction of future extremum (particularly relevant in inventory optimisation problems). Related to this issue, every time-series problem needs to be dealt with customised feature engineering technique for the sequential input data instead of exploiting the transfer learning capabilities prevalent in the domains of language processing or computer vision. In this regard, this paper makes the following major contributions to fill the gaps in research.

- The multi-attention feature embedding (MFE) network enabled by two novel elements, i.e. Score Normalised Attention (SNA) and Exponentially Decaying Attention (EDA) that generalise the ideas of Bahdanau and Luong attentions [3, 4].
- Use the feature embedding technique with transfer learning capability to achieve benchmark performance for multiple tasks such as forecasting, classification and extremum prediction.
- Extensive benchmarking of the simulation results with state of the art models such as N-BEATS [5], Temporal-Fusion-Transformer [2] and several other promising techniques, to demonstrate competitive model performance.

II. Background Work

Multi-horizon time series forecasting evolved rapidly with the invention of different recurrent layers aiding sequence-to-sequence forecasting. However, the Recurrent Neural Networks (RNN) suffer from vanishing gradient problem, as a solution to which frameworks like GRU and LSTM have been proposed. As an example, the canonical structure of [6] proposes an input sequence being mapped into a fixed dimensional state vector, and the encoded state vector gives a *context* to forecast the output sequence from an initial state.

However, [6] also argued that a single state representation obtained from LSTM or GRU caannot capture longer term dependencies in a multi-variate time series. The attention and transformer architectures evolved to overcome the limitation. The attention framework depends on the decoder's ability to attend to relevant parts of an input sequence ensuring that most important state vectors will be considered with the highest weights. The context can be mapped into an output by a dense neural network. The attention values (forming a Markov matrix) signify the weight of past tokens or time-steps to forecast a future token or time-step.

Hence, the attention mechanism is also considered as an information retrieval mechanism [7] with multiple keys, queries

and values. hence, it is equivalent to learning f, ensuring that

$$\mathbf{c} = f(\mathbf{q}, \mathbf{k}, \mathbf{v}) \tag{1}$$

where $\mathbf{q}$, $\mathbf{c}$, $\mathbf{v}$ and $\mathbf{k}$ are the query vectors, context vectors, value vectors and key vectors respectively. The goal of the is to ensure f learns to minimise a loss function. Different attention functions (e.g. [1, 3, 4]) can be formulated by (1).

III. Problem Formulation

In spite of the wide array of attention methods available in the literature (some of which will be used as benchmarks in our experimentations later), the existing body of works is inadequate in addressing the general problem of modelling multi-resolution time series [8], where different covariates are sampled at irregular intervals. Following the example of [3], the context vectors in purely temporal attention is formulated as

$$\mathbf{c}_i = \sum_j a_{ij} \mathbf{h_j} \tag{2}$$

where j represents encoder output index for time step j, i represents decoder output index for a future time-step, $\mathbf{h_j}$ is the j-th state output from the encoder and a_{ij} is the quantitative impact of past time step j on future time step i. In the literature, the attention weights a_{ij} are typically trained (jointly with other layers) in a way so as to minimise loss functions over the training samples. However, (2) assumes that all the $\mathbf{h_j}$ are based on samples located at equidistant or synchronous sampling intervals, i.e. all the covariates were sampled every day, or week etc. This renders the existing attention techniques ineffective when this constraint is not respected for irregular time series such as [9, 10].

As another problem related to this, the literature lacks any general embedding technique for different time series as it is available in Natural Language Processing or vision domain. As a result, the goal of this work is to device a vector space embedding technique for asynchronously and irregularly sampled time series with dynamic (forecasting or extremum prediction) or static (classification) targets. We divide the covariates into three categories

- *Cumulative Covariates*: These dynamic covariates accumulate over time-intervals such as hour, day or week. Examples include sales quantities, traffic flow through a junction etc. all measured over finite time intervals.
- *Instance Based Covariates*: These are dynamic covariates that are sampled at discrete time instances, and vary continuously, but do not accumulate over time. Examples include temperature, humidity data from sensors, health data such as blood pressure etc.
- *Static Covariates*: These are static features in a time series that do not vary with time. Examples include attributes of a specific patient (e.g. gender, nationality etc.), attributes of a specific store (e.g. size of store, distance from closest competitor etc.)

The goal is to arrive at a feature embedding technique exploiting the concept of Luong attention that can take these different types of features and outputs an embedding vector.

IV. Attention Driven Feature Embedding

This paper aims to generalise the traditional attention to make it resolution-aware, as pointed out in Section II. The unified model architecture consisting of different blocks is shown in Fig. 1. The encoder block uses the well-known Gated Recurrent Unit (GRU) to encode the dynamic features into a sequence of states. However, unlike previous literature, this paper makes no assumption about the synchronisation of sampling among different dynamic features. To address the issue of asynchronous sampling (which adversely affects the performance in traditional attention mechanism, as shown in Section V later), this paper proposes two novel architectures to deal with cumulative and instance based covariates, as described next.

A. Attention for Cumulative Covariates

As discussed previously in Section II, cumulative time series demonstrate additive behaviour over disjoint time intervals, which qualifies them as renewal process [11]. In other words, with evolution of time, their values are non-negative and non-decreasing. Mathematically speaking, if t_1 and t_2 are non-overlapping but consecutive time intervals, $c(t)$ is the number of events (such as customer traffic) over an interval t, then

- $c(t_1)$ and $c(t_2)$ (both being random variables) are independent
- $\mathbb{E}\left[c(t_1 + t_2)\right] = \mathbb{E}\left[c(t_1)\right] + \mathbb{E}\left[c(t_2)\right]$

The distributions of $c(t_1)$ and $c(t_2)$ depend on specific domain details, and can be modelled by several statistical tools. However, in this paper, the goal is to featurise past T values over consecutive, non-overlapping intervals via the Score Normalised Attention (SNA) mechanism.

Assume the past T look-back values are given by $(t_1, \mathbf{h}_1), (t_2, \mathbf{h}_2), \cdots, (t_T, \mathbf{h}_T)$ where $t_i{}_{i=1}^{i=T}$ are disjoint, consecutive time-intervals, and $\mathbf{h}_i{}_{i=1}^{i=T}$ are states of dynamic covariates[1].

1) Review of Traditional Attention: The standard practice in traditional attention mechanisms ([4], as an example) typically discard the $t_i{}_{i=1}^{i=T}$ values (assuming all of them are equally spaced) and encode the state values $\mathbf{h}_i{}_{i=1}^{i=T}$ into a context vector as follows.

$$\mathbf{c}_i = \sum_{j=1}^{j=T} a_{ij} \mathbf{h_j} \quad \forall 1 \leq i \leq \mathcal{T} \tag{3}$$

where $\mathbf{c}_i$ represents the context vector at i-th future time-step to be predicted, j represents the look-back time step and a_{ij} represent the *attention scores*, which is a measure of how much state value at look back step j influence the state value at future step i.

[1]The states may represent the raw values of the covariates, or the raw values passed through recurrent/LSTM networks [6].

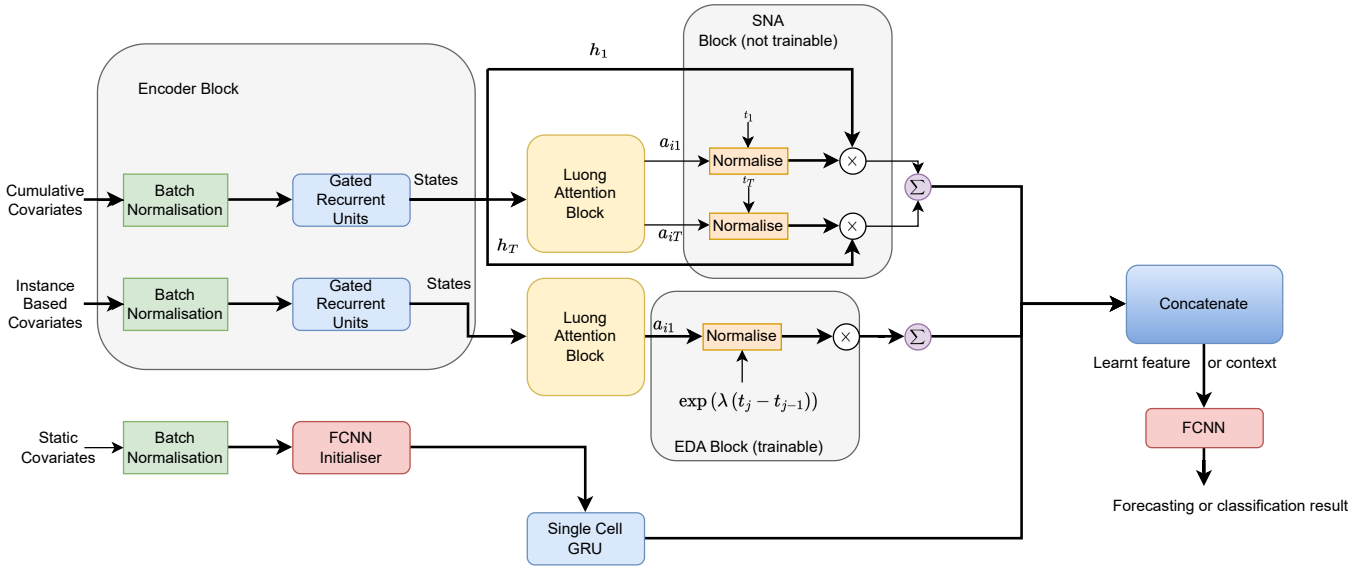

Fig. 1. Unified Model Architecture Combining SNA and EDA

B. Score Normalised Attention

Instead of discarding the lengths of time intervals $t_j{}_{j=1}^{i=T}$, this work proposes normalising the attention scores a_{ij} with by the corresponding time interval. As a result, the newly normalised attention score are given by

$$a_{ij}^{\text{norm}} = \frac{a_{ij}}{t_j} \quad \forall 1 \leq i \leq \mathcal{T}, 1 \leq j \leq T \tag{4}$$

Based on this, (3) can be modified to incorporate the new attention weights as

$$\mathbf{c}_i = \sum_{j=1}^{j=T} a_{ij}^{\text{norm}} \mathbf{h_j} \quad \forall 1 \leq i \leq \mathcal{T} \tag{5}$$

To be noted, this attention score normalisation is different from typical batch normalisation [12] in deep learning which normalises feature values to fit within the same scale. The major advantage of SNA is, it can exploit existing deep learning frameworks (tensorflow, for example) for jointly training the (unnormalised) attention weights. It works by appending a normalisation layer depending on the values of $t_j{}_{j=1}^{i=T}$ after the established Bahdanau or Luong attention [3, 4] methods. As the values of $t_j{}_{j=1}^{i=T}$ are directly available as part of training data or inference feature, no further processing is necessary other than carrying out the normalisation of (4). Complexity of the normalisation operation is $\mathcal{O}(T)$. The mechanism is described in Fig. 1.

C. Exponentially Decaying Attention

Instance based covariates do not subject themselves to the rules of counting process as discussed in Section IV-B. As they are (irregularly) sampled on different instances, as opposed to being measured on an interval, it is not possible to normalise the attention scores by the interval lengths. Hence, to featurise irregularly sampled instance based covariates, this paper proposes the Exponentially Decaying Attention (EDA) to modify the vanilla attention in (3).

Similar to Section IV-A1, the covariates are given as $(t_1, \mathbf{h}_1), (t_2, \mathbf{h}_2), \cdots, (t_T, \mathbf{h}_T)$. But in this case, $t_i{}_{i=1}^{i=T}$ represent the instants (counted from some arbitrary reference instant or epoch[2]). However, assume they are instance based covariates, which implies the value is sampled at an instant, instead of accumulated over an interval.

As a result, the following exponential normalisation is applied on attention weights from (3).

$$a_{ij}^{\text{norm}} = \frac{a_{ij}}{\exp(\lambda(t_{j+1} - t_j))} \quad \forall 1 \leq i \leq \mathcal{T}, 1 \leq j \leq T \tag{6}$$

The proposed Exponentially Decayed Attention (EDA) of (6) penalises data points that are further away along the look-back line with an exponential decay along the time axis. The trainable parameter $\lambda > 0$ represents the decay rate, which is fine-tuned via back-propagation [14] of errors, similar to other co-efficients in a neural network. The EDA block in Fig. 1 demonstrates the proposed decayed attention method. Note that, in contrast to SNA, the EDA introduces another additional trainable parameter λ which is the decay rate of attention for instance based covariates, and should be learnt from the training samples. Similar to SNA, complexity of the decay operation is $\mathcal{O}(T)$.

V. Experimental Results

This paper considered five datasets to compare the performance of the proposed embedding techniques against multiple benchmarks from recent literature on time-series-forecasting.

A. Datasets

1) Favorita Grocery Sales: This grocery items sales dataset from Favorita corporation comes from multiple provinces in

[2]An example is the widely used Unix epoch [13].

Ecuador. Each province is considered a class for the classification problems, and for forecasting problems, the objective is to predict the sales of different items.

2) San Francisco Traffic: The (normalised) road occupandy on an hourly basis was captured by 963 sensors San Francisco. We consider only five sensors, each as a separate class, which provide enough data for convergence and validation of the proposed approach.

3) Portugal Electricity Consumption: This shows hourly electricity load from households in Portugal. Similar to Section V-A2, five households are considered to demonstrate the effectiveness of the algorithm.

4) Physionet Challenge: The mortality data for hospital ICU patients was published in [9]. For classification purpose, the mortality of the patients is used as a binary target (1 means survival, 0 means diseased). For time series forecasting and extremum prediction, the temporal data is used.

5) Human Activity Dataset: The human activity dataset [10] was collected from a group of volunteers, each performing one of six different activities wearing a sensor-enabled smartphone on the waist. The sensor signals (accelerometer and gyroscope) are recorded together with the video recordings of each activity for manual labelling.

B. Metrics for the Categories of Problems

As the experimentations were conducted on different downstream tasks based on a similar feature-learning architecture, it calls for suitable metrics according to the problem class. This section summarises the metrics with relevant justifications.

1) Forecasting: As forecasting involves prediction of a wide range of output values of the time-series, it is imperative to *normalise* the errors (difference between prediction and ground truth) with the actual values. Hence, the Mean Absolute Percentage Error (MAPE) has been one of the widely used metric in recent literature [15] and used in this paper as well.

2) Extremum Prediction: The extremum prediction is tackled within next $\mathcal{T}$ time-steps for each dataset[3]. As the variance of outcome here is limited to $\mathcal{T}$ by definition, no normalisation is deemed necessary here. Hence, the Mean Absolute Error (MAE) is used as a metric.

3) Classification: For multi-class classification problem, accuracy has been the simplest metric to judge a model. However, it fails particularly when the classes are imbalanced, and accuracy metric cannot account for performance variation of the model on different classes. Hence, this paper uses the micro-F1 score [16] as a metric which weighs the individual class' F1 scores by the class counts.

C. Forecasting of Future Values

The following models were chosen from recent literature as benchmarks for future value forecasting.

- *N-BEATS*: The neural basis expansion technique by passing input features through a fork architecture and residual tracking was proposed in [5].
- *TFT*: The temporal fusion transformers was proposed in [2] to account for exogenous covariates in multi-horizon forecasting, as an extension of the original multi-head attention [1], providing interpretable attention weights.
- *SeFT*: The use of set functions for time series modelling, discarding the individual time stamps, was proposed in [17]. The approach is computationally parallelisable and has an efficient memory requirement, making it a good candidate for benchmarking in our experiments.
- *Informer*: The prob sparse attention for long sequence forecasting was published in [18] as a suitable alternative to self-attention for sparsely populated series with insufficient training data.
- *Perceiver*: This combination of latent transformers and cross-attention layers was proposed in [19], demonstrating competitive performance in multiple datasets.

The experiments were performed on a 64 bit Ubuntu machine running with 16-Core i9 processors and NVIDIA GeForce RTX 3050 GPU. The models were implemented with tensorflow version 2.10.0.

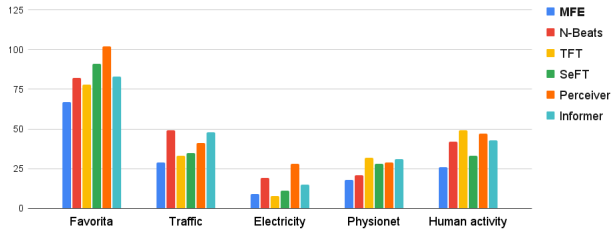

Fig. 2. Comparison of MAPE across Models

For comparison of forecast values against ground truth, the Mean Absolute Percentage Error (MAPE) is used for loss function. Fig. 2. Table I show the MAPE achieved by the benchmark models together with MFE. Table I also shows the properties of the datasets such as how many different classes (such as province, in case of Favourita, or number of traffic junctions) are considered (N), the look-back time steps (T) and the number of future steps ($\mathcal{T}$) considered for forecasting and extremum prediction. The overall competitive advantage of MFE to featurise the cumulative and instance based covariates is evidenced from the figure.

Most existing benchmarks such as Informer and N-BEATS do not account for differing resolutions on cumulative and instance based covariates, which adversely affect the prediction performance. Particularly, the Informer model [18] is based on prob-sparse attention that produces accurate result for sparse time series with efficient computational complexity. The experiment results show inadequacy of benchmark methods to model asynchronously sampled covariates.

D. Downstream Tasks based on Learnt Representation

The idea of MFE (enabled by SNA and EDA) also enable us to perform other downstream tasks based on the representation learnt from the raw features consisting of covariates,

[3]The $\mathcal{T}$ values can be read from Table I.

TABLE I
MAPE SCORES (IN PERCENTAGE) FOR FORECASTING OF FUTURE VALUES

Name	N	T	$\mathcal{T}$	**MFE**	N-BEATS	TFT	SeFT	Perceiver	Informer	Without SNA	Without EDA
Favorita	15	120	30	**67**	82	78	91	102	83	112	128
Traffic	5	192	24	**29**	49	33	35	41	48	74	61
Electricity	5	192	24	9	19	**8**	11	28	15	26	22
Physionet	2	15	5	**18**	21	32	28	29	31	54	39
Human activity	6	15	5	**26**	42	49	33	47	43	56	61

TABLE II
DOWNSTREAM TASKS WITH MFE AND COMPARISON WITH BENCHMARKS

Dataset and Downstream Tasks	**MFE**	Cat-Boost	XGBoost	LGBM	LDA	DCNN	Without SNA	Without EDA
MAE for Prediction of Maximum								
Favorita	**12.84**	18.36	21.3	15.04	21.02	16.94	21.02	32.05
Traffic	**9.47**	18.38	20.54	14.95	28.7	15.94	30.94	28.01
Electricity	8.38	**8.04**	19.24	11.02	12.93	9.31	12.39	19.01
Physionet	8.32	**7.91**	10.48	13.75	11.38	9.18	19.06	23.4
Human activity	**12.47**	18.04	13.91	14.9	21.28	19.17	18.47	21.09
MAE for Prediction of Minimum								
Favorita	**11.29**	12.04	18.73	21.03	18.02	12.38	18.02	17.9
Traffic	6.01	**5.29**	9.37	13.19	17.04	11.06	19.93	15.38
Electricity	**7.52**	9.23	11.98	10.46	12.04	15.02	29.83	18.05
Physionet	9.27	**8.18**	19.19	12.05	21.04	16.94	18.01	12.04
Human activity	**10.02**	11.93	20.91	16.47	15.49	18.5	21.48	19.03
Micro-F1 Scores for Classification (in percentage)								
Favorita	**11.29**	12.04	18.73	21.03	18.02	12.38	15.6	17.04
Traffic	6.01	**5.29**	9.37	13.19	17.04	11.02	13.33	12.48
Electricity	**7.52**	9.23	11.98	10.46	12.04	15.02	19.57	26.40
Physionet	9.27	**8.18**	19.19	12.05	21.04	16.94	24.05	21.03
Human activity	**10.02**	11.93	20.91	16.47	15.49	18.5	16.49	19.03

basically exploiting the idea of transfer learning. The three other downstream tasks targeted in this paper are

- Prediction of maximum and minimum in the future
- Classification of the specific individual time series (such as which patient does a series represent, or which traffic junction).

As tree based learning models have been more competitive in literature in such tasks, the following combination of tree based boosting models, discriminant analyses based models, and convolutional network were chosen as benchmarks for the tasks of classification as well as extremum prediction.

- *CatBoost*: The ordered boosting and novel encoding of categorical features was introduced by [20] and since then became a benchmark model for supervised classification and regression problems.
- *XGBoost*: The sparsity aware end-to-end tree boosting for supervised learning problem was proposed in [21] and is widely used for supervised prediction at massive scale using innovative cache access patterns.
- *LGBM*: The gradient based sampling and exclusive feature bundling to further improve prediction efficiency was proposed in [22].
- *LDA*: Linear discriminant analysis has been a classic probability theory based method applied on massive data sets [23], assuming a Gaussian distribution of target when conditioned upon the input feature.
- *DCNN*: Dilated convolutions have been an important technique for spatial pattern detection in images, and [24] extended the concept to time series featurisation using temporal patterns.

The benchmarking results for three downstream tasks against the five datasets, are presented in Table II. The advantage of representation learning via MFE is apparent from the table. T he classical tree boosting methods use

a combination of lag features (for dynamic covariates) and static covariates for modelling. However, they cannot usually account for the asynchronous sampling of dynamic covariates, which is the major contribution of this paper for featurisation of these variables. However, even among the classical models, the Category Boosting has shown significant promise dealing with problems involving multiple static covariates. This is evidenced in Table II where CatBoost provides a competitive performance.

E. Ablation Study

The experiments also included an ablation study of removing the proposed SNA and EDA mechanisms (separately), replacing them with traditional Luong attention layers which do not account for the different resolution and asynchronous nature of the time series. The results of the ablation analysis for forecasting and downstream tasks are also presented in the final two columns of Tables I and II. It is seen that the performances for downstream tasks are significantly impacted by the absence of SNA and EDA layers. For future value forecasting, the absence of EDA is more critical for Favorita dataset than absence of SNA. This is due to the presence of fewer instance based covariates in the Favorita dataset, and most feature variables are cumulative covariates. This is also observed for the human activity dataset. For the other dataset (with more instance based covariates) show better performance without EDA than without SNA.

VI. Conclusions& the Way Ahead

The paper proposed the novel multi-attention feature embedding algorithm generalising Bahdanau and Luong attention. In addition to forecasting of future values, the feature embedding techniques enable exploiting transfer learning in time series (similar to BART in natural language processing domain) to handle distinct downstream tasks such as classification and forecasting. The proposed algorithm allows for a joint training with the downstream DNN layers, and the benchmarking shows competitive performance against relevant state-of-the-art methods in literature. Further work will focus on more distributed and parallelised learning algorithms for feature embedding.

Acknowledgement

This research is supported by A*STAR under Supply Chain (SC) 4.0 Digital Supply Chain Development via Platform Technologies Programme (Grant No. M21J6a0080). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of A*STAR.

References

[1] 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*, Long Beach, 2017, pp. 5999–6009.

[2] B. Lim, S. Ark, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[3] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, San Diego, 2015.

[4] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Conference on Empirical Methods in Natural Language Processing*, Lisbon, 2015, pp. 1412–1421.

[5] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations*, New Orleans, 2019.

[6] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoderdecoder approaches," in *8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, 2014, pp. 103–111.

[7] B. J. Jansen and S. Y. Rieh, "The Seventeen Theoretical Constructs of Information Searching and Information Retrieval," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 8, pp. 1517–1534, 2010.

[8] X. Wu, B. Shi, Y. Dong, C. Huang, L. Faust, and N. V. Chawla, "RESTFul: Resolution-aware forecasting of behavioral time series data," in *International Conference on Information and Knowledge Management*. Torino, Italy: Association for Computing Machinery, 2018, pp. 1073–1082.

[9] I. Silva, G. Moody, D. Scott, L. Celi, and R. Mark, "Predicting in-hospital Mortality of ICU Patients: The Physionet/Computing in Cardiology Challenge," *Computing in Cardiology*, vol. 39, pp. 245–248, 2012.

[10] "Human Activity Dataset." [Online]. Available: https://tinyurl.com/2245vpng

[11] P. Brémaud, *Probability theory and stochastic processes*. New York: Springer International Publishing, 2020.

[12] Z. Liao and G. Carneiro, "On the importance of normalisation layers in deep learning with piecewise linear activation units," in *IEEE Winter Conference on Applications of Computer Vision*, Lake Placid, 2016.

[13] *Unix Programmer's Manual*. Bell Labs, 1971.

[14] D. E. Rumelhard, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, pp. 533–536, 1986.

[15] S. Barman Roy, M. Yuan, Y. Fang, and M. K. Sett, "Spatio-Temporal Attention with Symmetric Kernels for Multivariate Time Series Forecasting," in *IEEE Conference on Industrial Electronics and Applications*, Chengdu, 2022.

[16] D. Harbecke, Y. Chen, L. Hennig, and C. Alt, "Why only Micro-F1? Class Weighting of Measures for Relation Classification," in *Workshop on Efficient Benchmarking in Natural Language Processing*, 2022, pp. 32–41.

[17] M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt, "Set functions for time series," in *International Conference on Machine Learning*, 2020, pp. 4303–4313.

[18] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *Proceedings of Association for the Advancement of Artificial Intelligence*, 2021.

[19] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General Perception with Iterative Attention," in *International conference on machine learning*, 2021, pp. 4651–4664.

[20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, Montreal, 2018, pp. 6638–6648.

[21] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 2016, pp. 785–794.

[22] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, Long Beach, 2017, pp. 3147–3155.

[23] E. K. Ngailo, "Contributions to linear discriminant analysis with applications to growth curves," Ph.D. dissertation, Linköping University, Linköping, 2020.

[24] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *IEEE Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.