A Survey on Transformer-based Framework for Multivariate Time Series Representation Learning

Rahul Mangalampalli, Keerthan Nandigama

ECE Department, Stony Brook University
Stony Brook, New York, USA

rahul.mangalampalli@stonybrook.edu | keerthan.nandigama@stonybrook.edu

`

ABSTRACT

In our exploration, we reproduced an innovative framework designed for the learning of multivariate time series representations, rooted in the transformer encoder architecture. This framework incorporates an unsupervised pre-training approach, exhibiting notable performance advantages compared to fully supervised learning in subsequent tasks. Remarkably, these benefits persist even in scenarios where additional unlabeled data is not leveraged, relying solely on the existing data samples. In our experimentation, we applied this framework to three publicly available multivariate time series datasets spanning diverse domains and characteristics. The outcomes underscore its comparable efficacy in classification tasks relative to the presently supervised methodologies. Notably, this holds true even for datasets with limited training samples, sometimes as few as a few hundred. Our decision to incorporate this framework in our experimentation stems from its ability to surpass the state-of-the-art, marking a significant breakthrough in unsupervised learning.

CCS CONCEPTS

• Computing methodologies → Unsupervised learning; Supervised learning; Neural networks.

KEYWORDS

transformer; deep learning; multivariate time series; unsupervised learning; self-supervised learning framework; regression; classification; imputation

1. INTRODUCTION

Multivariate time series (MTS) represents a crucial data form in diverse fields such as science, medicine, finance, engineering, and industrial applications [2]. It captures the evolution of synchronized variables, depicting simultaneous measurements of distinct physical quantities over time. Despite the increasing prevalence of MTS data in the realm of "Big Data," labeled data remains limited, posing challenges due to the associated costs and impracticality of extensive labeling efforts [2]. This has sparked interest in methodologies that promise high accuracy with either a limited amount of labeled data or by leveraging existing unlabeled data [30].

In the domain of time series modeling, particularly in forecasting, regression, and classification tasks, deep learning models, including InceptionTime and ResNet, compete with non-deep learning counterparts such as TS-CHIEF, HIVE-COTE, and ROCKET, which currently excel in time series regression and classification benchmarks [8]. This work introduces a transformer encoder for unsupervised representation learning of multivariate time series, extending its application to time series regression and classification [27]. Transformers, initially designed for natural language translation, have demonstrated state-of-the-art performance in Natural Language Processing (NLP) tasks, excelling in various domains such as polyphonic music composition [15]. The unique multi-headed attention mechanism of transformers, particularly suited for time series data, allows simultaneous representation of each input sequence element by considering its context in both past and future. Inspired by the success of unsupervised pre-training in NLP, this work proposes a universally applicable methodology leveraging unlabeled data [5, 10]. The approach involves training a transformer encoder using an input "denoising" (autoregressive) objective to derive dense vector representations of multivariate time series [27]. The pre-trained model proves effective in diverse downstream tasks, including regression, classification, imputation, and forecasting. Applying this framework to public datasets for multivariate time series regression and classification tasks reveals superior performance compared to current state-of-the-art approaches [12], even with a limited amount of training data samples. Importantly, this work highlights the advantages of unsupervised learning over supervised learning for classification and regression of multivariate time series, even without additional unlabeled data samples [2].

It is noteworthy that our reproduction of the paper with tailored parameters demonstrates the efficiency of our models, featuring at most hundreds of thousands of parameters. This efficiency allows training comparable in speed to lean non-deep learning-based approaches, utilizing commodity GPUs.

2. RELATED WORK

Presently, state-of-the-art methods for the regression and classification of time series data primarily revolve around non-deep learning techniques, including TS-CHIEF [29], HIVE-COTE [21], and ROCKET [8, 9]. These methodologies have been established through evaluations on public benchmarks [2, 30]. Following this, deep learning approaches, specifically CNN-based architectures such as InceptionTime [12] and ResNet [11], also contribute to the landscape. ROCKET, recognized as the top-ranking method on average, adopts a unique strategy by training a linear classifier on features extracted from a diverse set of random convolutional kernels. This method excels in capturing varied temporal patterns within the data. A related development to our own work is MiniROCKET [9], a variant of ROCKET designed to enhance processing efficiency while maintaining comparable accuracy levels.

In contrast, HIVE-COTE and TS-CHIEF (inspired by Proximity Forest [23]) represent sophisticated methodologies that integrate expert insights into time series data. They utilize large, heterogeneous ensembles of classifiers, incorporating techniques such as shapelet transformations, elastic similarity measures, spectral features, and random interval and dictionary-based approaches. Despite their complexity and ability to leverage expert domain knowledge, these methods come with significant computational costs, lack compatibility with GPU hardware, and exhibit poor scalability when dealing with datasets featuring numerous samples and long time series. Additionally, it's worth noting that these methods have been developed for and evaluated exclusively on univariate time series data.

Exploration of Unsupervised Learning for Multivariate Time Series:

Recent endeavors in the realm of unsupervised learning for multivariate time series have predominantly centered around the application of autoencoders. These autoencoders, primarily implemented as either Multi-Layer Perceptrons, with a focus on tasks like clustering and visualizing shifting sample topology over time [13, 17], or as RNN (most commonly LSTM) sequence-to-sequence networks [24, 26], have been trained with an input reconstruction objective.

Taking a distinctive approach, Bianchi et al. [4] introduce a novel autoencoding method to address missing data. They employ a stacked bidirectional RNN encoder and stacked RNN decoder, leveraging a user-provided kernel matrix as prior information to condition internal representations. This approach aims to encourage the learning of similarity-preserving representations of the input.In the context of time series clustering, Lei et al. [18] adopt a method focused on preserving similarity between time series. They direct learned representations to approximate a distance metric, such as Dynamic Time Warping (DTW), between time series through a matrix factorization algorithm. Zhang et al. [34] deviate by employing a composite convolutional-LSTM network with attention. Their approach includes a loss function that targets the reconstruction of correlation matrices between variables in the multivariate time series input. Notably, this method is evaluated solely for the task of anomaly detection. Jansen et al. [16] introduce a distinctive approach relying on a triplet loss and the concept of temporal proximity. The loss function rewards similarity of representations between proximal segments while penalizing similarity between distal segments of the time series. Franceschi et al. [14] build on this idea by combining the triplet loss with a deep causal CNN with dilation, making the method effective for very long time series. Although outperformed by supervised state-of-the-art methods in univariate classification, it was, prior to our method, the leading unsupervised learning approach for univariate and multivariate classification datasets of the UEA/UCR archive [2].

Transformer Models for Time Series:

In a distinct avenue of exploration, recent studies have employed a full encoder-decoder transformer architecture for univariate time series forecasting. Li et al. [19] demonstrated superior performance compared to classical statistical methods such as ARIMA and recent models like TRMF, DeepAR, and DeepState. Wu et al. [33] utilized a transformer for forecasting influenza prevalence, showcasing advantages over ARIMA, an LSTM, and a GRU Seq2Seq model with attention. Lim et al. [20] applied a transformer for multi-horizon univariate forecasting, providing support for the interpretation of temporal dynamics. Furthermore, [25] employed an encoder-decoder architecture with a variant of self-attention for imputation of missing values in multivariate, geo-tagged time series, outperforming both classic and state-of-the-art RNN-based imputation methods on various datasets. In contrast to these approaches, our work aims to extend the utility of transformers beyond specific generative tasks requiring the full encoder-decoder architecture. We aspire to establish a broader framework that allows for unsupervised pre-training and can be readily adapted for a wide array of downstream tasks by modifying the output layer. This approach draws parallels to the transformative impact of BERT [10] in converting a language translation model into a versatile framework based on unsupervised learning, a methodology that has become a de facto standard, establishing the dominance of transformers in NLP.

3. METHODOLOGY

3.1 Base Model:

The method's foundation rests upon a transformer encoder, as detailed in the original work by Vaswani et al. [32], with a deliberate omission of the decoder segment in the architecture. This choice emanates from recognizing that the decoder module is well-suited for generative tasks, particularly in scenarios where there's no predefined output sequence length, such as translation, summarization in NLP, or forecasting in time series. However, the decoder module relies on the (masked) "ground truth" output sequence as input, rendering it unsuitable for tasks like classification or (extrinsic) regression. In contrast, the aim of this work is to formulate a unified framework adaptable to a diverse array of tasks. An architecture featuring only an encoder proves to be versatile, handling tasks like classification, regression, imputation, and generative tasks such as forecasting. The decision to use only an encoder also allows for the utilization of approximately half the model parameters, resulting in computational and learning benefits, including the avoidance of overfitting. Figure 1 illustrates a schematic diagram of the generic part of the model, consistent across all considered tasks. For an in-depth understanding of the transformer model, readers are directed to the original work. Here, the focus is on presenting the proposed changes that render it compatible with multivariate time series data rather than sequences of discrete word indices.

In this adaptation, each training sample $X \in \mathbb{R}^{w \times m} = [x_1, x_2, \dots, x_w]$. The original feature vectors x_t undergo normalization, with each dimension adjusted by subtracting the mean and dividing by the variance across training set samples. Subsequently, these vectors are linearly projected onto a d-dimensional vector space where d is the dimension of the transformer model sequence element representations. The linear projection is expressed by the equation:

$$u_t = W_p x_t + b_p \qquad (1)$$

Where $W_p \in \mathbb{R}^{d \times m}$, $b_p \in \mathbb{R}^d$, and $u_t \in \mathbb{R}^d$, $t = 0, \dots . w$ are the model input vectors, analogous to the word vectors of the NLP transformer. These vectors then serve as queries, keys, and values in the self-attention layer after adding positional encodings and multiplying by the corresponding matrices. Ultimately, as the transformer operates as a feed-forward architecture inherently indifferent to the input's ordering, an initiative is taken to imbue it with an awareness of the sequential nature inherent in time series data. This is accomplished by introducing positional encodings $W_{pos} \in \mathbb{R}^{w \times d}$ to the input vectors $U \in \mathbb{R}^{w \times d} = [u_1, u_2, \dots, u_w]$: $U' = U + W_{pos}$.

In lieu of employing predetermined sinusoidal encodings, an alternative approach is taken by incorporating fully learnable positional encodings. This deviation stems from our observation that these encodings exhibit superior performance across all datasets examined in this study. Evaluating the performance of our models, we note that the positional encodings generally do not significantly disrupt the numerical information within the time series. This observation mirrors the behavior seen in word embeddings, suggesting a hypothesis that these encodings are learned to inhabit a distinct, approximately orthogonal subspace compared to the one containing the projected time series samples. This approximate orthogonality condition is more attainable in high-dimensional spaces.

An essential consideration for time series data involves potential variations in the length of individual samples. Our framework adeptly addresses this concern: by establishing a maximum sequence length$w$for the entire dataset, shorter samples are padded with arbitrary values. Simultaneously, a padding mask is generated, introducing a substantial negative value to the attention scores for the padded positions before computing the self-attention distribution with the softmax function. This design ensures the model systematically disregards padded positions, enabling the parallel processing of samples in large minibatches. Contrary to the common practice in NLP transformers, which employs layer normalization after computing self-attention and after the feed-forward component of each encoder block, we opt for batch normalization. This choice is made to alleviate the impact of outlier values in time series, a concern not encountered in the context of NLP word embeddings. Additionally, we acknowledge the comparatively subpar performance of batch normalization in certain scenarios.

3.2 Classification:

The foundational model architecture introduced in Section 3.1 and illustrated in Figure 1 can be employed for the purposes of classification with the following modification: the final representation vectors $z_t \in \mathbb{R}^d$ corresponding to all time steps are concatenated into a single vector $\bar{z} \in \mathbb{R}^{d.w} = [z_1; \dots .; z_w]$. This concatenated vector $\bar{z}$ then serves as the input to a linear output layer with parameters $W_o \in \mathbb{R}^{n \times (d.w)}$, where $b_o \in \mathbb{R}^n$ is the number of classes for the classification problem.

$$\hat{y} = W_o \bar{z} + b_o \qquad (3)$$

For classification, the predicted values $\bar{\hat{y}}$ undergo an additional step where they are passed through a softmax function to derive a distribution across classes. The sample loss is then calculated as the cross-entropy between this distribution and the categorical ground truth labels.

During the fine-tuning of pre-trained models, we adopt an approach where training encompasses all weights. In contrast, freezing all layers except for the output layer would equate to utilizing static, pre-extracted representations of the time series.

4. UNSUPERVISED PRE-TRAINING

For the unsupervised pre-training phase of our model, we adopt the autoregressive task focused on denoising the input. Specifically, we deliberately introduce noise by setting a portion of the input to 0 and tasking the model with predicting the values of the masked elements. The schematic representation of this setup is illustrated in the right section of Figure 1.

To achieve this, we create a binary noise mask, denoted as $M \in \mathbb{R}^{w \times m}$, independently for each training sample and epoch. The input undergoes masking through elementwise multiplication: $\bar{X} = M \cdot X$. On average, a proportion r of each mask column, corresponding to a single variable in the multivariate time series, is set to 0 by alternating between segments of 0s and 1s.

We carefully choose state transition probabilities such that each masked segment (a sequence of 0s) follows a geometric distribution with a mean $\boldsymbol{l_m}$ and is succeeded by an unmasked segment (a sequence of 1s) with a mean length l_u= (1−r/r) l_m. We set $l_m = 3$ for all presented experiments. The rationale behind controlling the length of the masked sequence, as opposed to employing a Bernoulli distribution with parameter r for setting all mask elements independently at random, is to avoid the trivial prediction of very short masked sequences.

In our experiments, we found $r = 0.15$ to be effective, and we consistently use this value. This masking approach differs from the "cloze type" masking utilized by NLP models like BERT. Instead of replacing the original word embedding with a special token, our method encourages the model to attend to both preceding and succeeding segments in individual variables and to contemporaneous values of other variables in the time series. This

promotes the learning of inter-dependencies between variables.

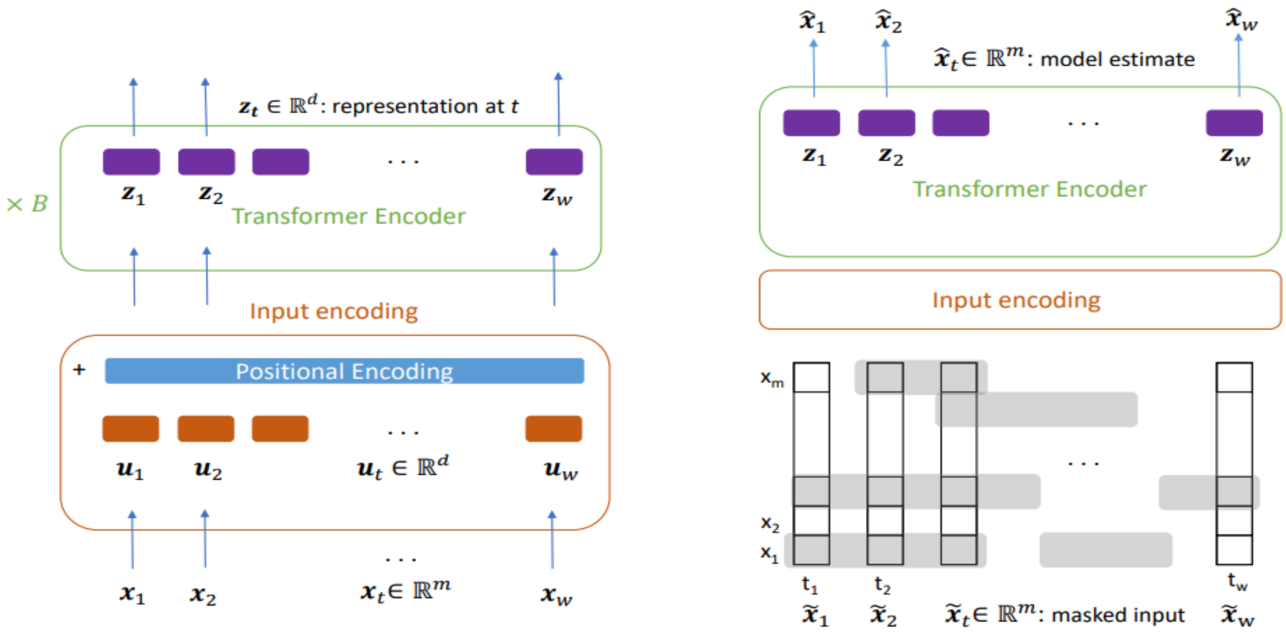


Figure 1: Left: Generic model architecture, common to all tasks. The feature vector xt at each time step t is linearly projected to a vector u_t of the same dimensionality d as the internal representation vectors of the model and is fed to the first self-attention layer to form the keys, queries and values after adding a positional encoding. Right: Training setup of the unsupervised pretraining task. We mask a proportion r of each variable sequence in the input independently, such that across each variable, time segments of mean length lm are masked, each followed by an unmasked segment of mean length l_u= (1−r/r) l_m. Using a linear layer on top of the final vector representations z_t , at each time step the model tries to predict the full, uncorrupted input vectors z_t; however, only the predictions on the masked values are considered in the Mean Squared Error loss.

For the denoising task, we utilize a linear layer with parameters $W_o \in \mathbb{R}^{m\times d}$ an $b_o \in \mathbb{R}^m$ on top of the final vector representations $z_t \in \mathbb{R}^d$. At each time step, the model simultaneously outputs its estimate $\hat{x}_t$ of the full, uncorrupted input vectors x_t. However, only predictions on the masked values, with indices in the set $M \equiv \{(t,i): m_{t,i} = 0\}$, are considered in the Mean Squared Error loss for each data sample:

$$\hat{x}_t = W_o z_t + b_o \tag{4}$$

$$LMSE = \frac{1}{|M|}\sum_{(t,i)\in M} (\hat{x}(t,i) - x(t,i)) \tag{5}$$

This objective differs from the one used by denoising autoencoders, where the loss considers the reconstruction of the entire input under typically Gaussian noise corruption. Additionally, our approach distinguishes itself from simple dropout on input embeddings, both in terms of statistical distributions of masked values and the fact that the masks also determine the loss function. Furthermore, we incorporate a 10% dropout when training all our supervised and unsupervised models.

5. EXPERIMENTS AND RESULTS

5.1 Datasets:

1. Face Detection Dataset

Data for this study originates from a Kaggle competition in 2014. The task involves determining whether a subject viewed a face or a scrambled image based on MEG, independent of the subject. The dataset comprises training data split into 10 subjects (subject01 to subject10) and 6 test subjects (subject11 to 16), totaling 5890 train trials and 3524 test trials. Each trial consists of 1.5 seconds of MEG recording, down-sampled to 250Hz and high-pass filtered at 1Hz, yielding 62 observations per channel across 306 channels [1].

2. TiSeLaC

TiSeLaC is the Time Series Land Cover Classification Challenge for MSTC, focusing on land cover classification using data from the Reunion Island. A case corresponds to a pixel with measurements over 23 time points, including 7 surface reflectances and 3 indices. There are 9 land cover types, and the winning entry achieved an F score of 99.29. The data formatting was done by Gonzalo Martinez and Tony Bagnall at UEA [1].

3. JapaneseVowels Dataset.

The A UCI Archive dataset involves 9 Japanese-male speakers pronouncing the vowels 'a' and 'e.' Applying '12-degree linear prediction analysis' to raw recordings yields time-series with 12 dimensions, originally ranging from 7 to 29 in length, now padded to 29. The classification task is speaker prediction, with each instance as a transformed utterance of 12*29 values and a single class label ([1...9]). The training set has 30 utterances per speaker, while the test set varies (24 to 88 instances per speaker) due to external factors[1].

4. Sleep Dataset.

The Sleep dataset consists of 153 whole-night sleeping EEG recordings formatted for classification. Collected from 82 healthy subjects, the 1-lead EEG signal is sampled at 100 Hz, and cases are segmented into subseries, each labeled with one of five sleeping patterns. The split dataset comprises 371,055 train, 107,730 validation, and 90,315 test cases [1]. We randomly sampled 80,000 training and 16,000 testing samples due to resource constraints.

5.2 Experiments:

In the course of handling datasets, we divided the training set into two portions, with 80% dedicated to unsupervised pretraining and the remaining 20% reserved for validation purposes. Following this, the model underwent fine-tuning on the complete training set and was evaluated on the official test set. We initially configured the model's hyperparameters based on the original paper and applied this setup to train the FaceDetection dataset, yielding consistent results.

However, we encountered an issue during training where unsupervised loss exhibited a significant drop after the initial epoch, as depicted in Figure 2. To address this, we simplified the multivariate transformer model by reducing its parameters and determined the number of training epochs through a trial-and-error approach, as outlined in Table 1. The refined model configuration led to a smoother reduction in loss for the unsupervised model. Additionally, we adjusted the learning rate by decreasing it by a factor of 0.1 specifically during the training of the Sleep dataset, resulting in a more gradual decline in the loss as shown in the Figure 3. Further details on these adjustments and their outcomes can be found in the corresponding figures and tables mentioned earlier.

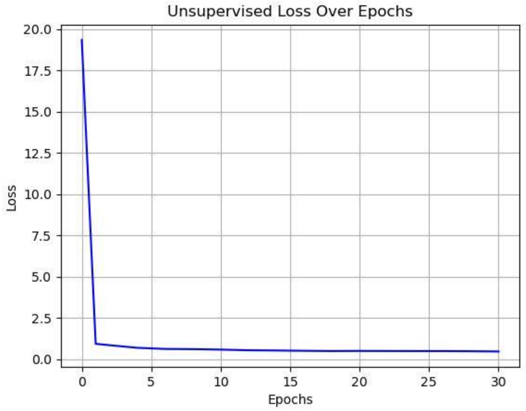


Figure 2: Unsupervised loss on the FaceDetection dataset before optimization.

When examining time series datasets with extended timeframes per data point, as seen in the sleep dataset, some interesting observations emerge. Despite a visible decrease in loss during training, the accuracy consistently stays around 0.5. A closer look revealed a key factor affecting this performance. The sleep dataset, known for its long timeframes, presents a unique challenge due to its univariate nature. This poses a challenge for the self-attention layer, highlighting the importance of considering dataset-specific features when using transformer models for time series tasks. This is especially crucial when dealing with univariate characteristics that might impact the effectiveness of certain layers.

Digging into this limitation, we experimented with five different transformer setups, including two lighter models, one with balanced settings, and two complex models. The results are summarized in Table 2 in the same sequence. Importantly, these

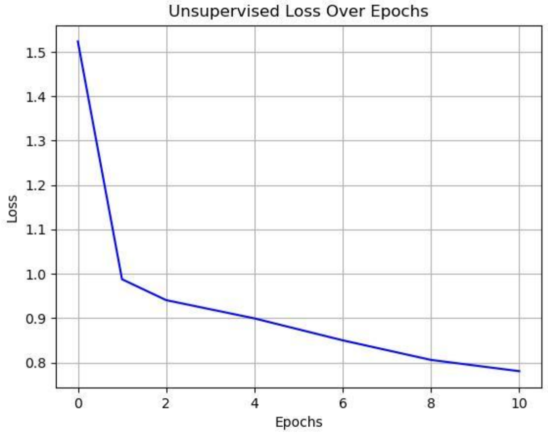


Figure 3: Unsupervised loss on the FaceDetection dataset after optimization.

Dataset	No. of Blocks	No. of Heads	Model Dimensions	FeedForward Dimensions	Unsupervised Loss	Finetune Loss	Supervised Loss	Finetune Accuracy	Supervised Accuracy
FaceDetection	1	1	16	32	0.78	0.62	0.63	0.67	0.66
TiSeLaC	1	1	16	32	0.093	0.42	0.39	0.86	0.87
JapaneseVowels	1	1	16	32	1.25	0.18	1.003	0.97	0.81
Sleep	1	1	32	64	0.97	1.20	1.21	0.54	0.53

Table 1: Hyperparameter configuration, final losses and accuracies after training for 10 epochs in unsupervised training paradigm, then finetuning for 5 epochs, and for 15 epochs in supervised training paradigm.

configurations clearly show that applying self-attention is disadvantageous for the univariate sleep dataset.

This challenge aligns with the broader discourse on adapting transformer architectures to diverse time series datasets. Studies such as [35] emphasize the need for tailoring models to dataset nuances, considering factors like dimensionality. Additionally, the exploration of transformer models in time series applications is discussed in [36], providing valuable insights into the potential challenges associated with datasets and the importance of refining architectural choices to enhance model performance.

Dataset	No. of Blocks	No. of Heads	Model Dimensions	FeedForward Dimensions	Learning Rate	Finetune Accuracy	Supervised Accuracy
1	1	1	8	8	0.001	0.507	0.513
2	1	1	16	32	0.001	0.49	0.52
3	1	1	32	64	0.001	0.54	0.53
4	1	1	256	512	0.0001	0.533	0.531
5	1	1	512	1024	0.0001	0.52	0.50

Table 2: Different model settings and their performance on the sleep

6. FUTURE WORK

It becomes essential to devise a methodology that allows for the downsampling of lengthier time series data to a manageable size, conducive to comprehension by the transformer model. Following this, there arises a need to delicately restore the processed data to its original configuration, reminiscent of an encoder-decoder setting, particularly during the unsupervised phase. This approach ensures the adaptability of the transformer model to extensive time series data while maintaining the fidelity of the information throughout the learning process.

7. CONCLUSION

Upon investigating the model's performance across four distinct datasets, a noteworthy revelation unfolded: the efficacy of a more compact transformer architecture in discerning intrinsic patterns within time series data. Impressively, this streamlined model exhibits efficiency in training, converging within a concise timeframe, typically around 15 epochs. Additionally, our exploration underscored the effectiveness of an unsupervised learning paradigm, proving advantageous in scenarios with sparse or absent labels. Intriguingly, the unsupervised approach demonstrated comparable performance to its supervised counterpart, showcasing its potential as a viable alternative in label-scarce situations.

ACKNOWLEDGEMENTS

We sincerely thank prof. Fan Ye for his constant support during this project. He insights and feedback gave us immense push to delve deeper in to this project. Additionally, we thank the original author of this paper, George Zerveas, for shedding light on an issue that we were facing during the training phase of the project.

REFERENCES

[1] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. arXiv:1811.00075 [cs, stat] (Oct. 2018).

[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. 2017. The Great Time Series Classification Bake Off: a Review and Experimental Evaluation of Recent Algorithmic Advances. Data Mining and Knowledge Discovery 31 (2017), 606– 660. Issue 3.

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The LongDocument Transformer. arXiv:2004.05150 [cs] (April 2020).

[4] Filippo Maria Bianchi, Lorenzo Livi, Karl Øyvind Mikalsen, Michael Kampffmeyer, and Robert Jenssen. 2019. Learning representations of multivariate time series with missing data. Pattern Recognition 96 (Dec. 2019), 106973. https://doi.org/ 10.1016/j.patcog.2019.106973

[5] T. Brown, B. Mann, et al. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. arXiv:1901.02860 [cs, stat] (June 2019).

[7] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. 2019. GRU-ODEBayes: Continuous Modeling of Sporadically-Observed Time Series. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 7379–7390.

[8] Angus Dempster, Franccois Petitjean, and Geoffrey I. Webb. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery (2020). https: //doi.org/10.1007/s10618-020-00701-z

[9] Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. 2020. MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification. arXiv:2012.08791 [cs, stat] (Dec. 2020).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805

[11] Hassan Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33, 4 (July 2019), 917–963. https://doi. org/10.1007/s10618-019-00619-1

[12] H. Fawaz, B. Lucas, et al. 2019. InceptionTime: Finding AlexNet for Time Series Classification. ArXiv (2019). https://doi.org/10.1007/s10618-020-00710-y

[13] Vincent Fortuin, M. Hüser, Francesco Locatello, Heiko Strathmann, and G. Rätsch. 2019. SOM-VAE: Interpretable Discrete Representation Learning on Time Series. ICLR (2019).

[14] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 4650– 4661.

[15] Cheng-Zhi Anna Huang, Ashish Vaswani, et al. 2018. Music transformer: Generating music with long-term structure. In International Conference on Learning Representations.

[16] A. Jansen, M. Plakal, Ratheet Pandya, D. Ellis, Shawn Hershey, Jiayang Liu, R. C. Moore, and R. A. Saurous. 2018. Unsupervised Learning of Semantic Audio Representations. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2018). https://doi.org/10.1109/ICASSP.2018.8461684

[17] A. Kopf, Vincent Fortuin, Vignesh Ram Somnath, and M. Claassen. 2019. Mixture of-Experts Variational Autoencoder for clustering and generating from similarity based representations. ICLR 2019 (2019).

[18] Qi Lei, Jinfeng Yi, R. Vaculín, Lingfei Wu, and I. Dhillon. 2017. Similarity Preserving Representation Learning for Time Series Analysis. ArXiv (2017).

[19] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In Advances in Neural Information Processing Systems. 5243–5253.

[20] Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. 2020. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. arXiv:1912.09363 [stat.ML]

[21] J. Lines, Sarah Taylor, and Anthony J. Bagnall. 2018. Time Series Classification with HIVE-COTE. ACM Trans. Knowl. Discov. Data (2018). https://doi.org/10. 1145/3182382

[22] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the Variance of the Adaptive Learning Rate and Beyond. arXiv:1908.03265 [cs, stat] (April 2020).

[23] Benjamin Lucas, Ahmed Shifaz, et al. 2019. Proximity Forest: An effective and scalable distance-based classifier for time series. Data Mining and Knowledge Discovery 33, 3 (May 2019), 607–635. https://doi.org/10.1007/s10618-019-00617-3

[24] Xinrui Lyu, Matthias Hueser, Stephanie L. Hyland, George Zerveas, and Gunnar Raetsch. 2018. Improving Clinical Predictions through Unsupervised Time Series Representation Learning. In Proceedings of the NeurIPS 2018 Workshop on Machine Learning for Health. arXiv:1812.00490

[25] J. Ma, Zheng Shou, Alireza Zareian, Hassan Mansour, A. Vetro, and S. Chang. 2019. CDSA: Cross-Dimensional Self-Attention for Multivariate, Geo-tagged Time Series Imputation. arXiv:1905.09904 [cs.CS]

[26] P. Malhotra, T. Vishnu, L. Vig, Puneet Agarwal, and G. Shroff. 2017. TimeNet: Pre-trained deep recurrent neural network for time series classification. ESANN (2017).

[27] Colin Raffel, Noam Shazeer, et al. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ArXiv abs/1910.10683 (2019).

[28] Sheng Shen, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. PowerNorm: Rethinking Batch Normalization in Transformers. arXiv:2003.07845 [cs] (June 2020).

[29] Ahmed Shifaz, Charlotte Pelletier, F. Petitjean, and Geoffrey I. Webb. 2020. TSCHIEF: a scalable and accurate

forest algorithm for time series classification. Data Mining and Knowledge Discovery (2020). https://doi.org/10.1007/s10618- 020-00679-8

[30] C. Tan, C. Bergmeir, François Petitjean, and Geoffrey I. Webb. 2020. Monash University, UEA, UCR Time Series Regression Archive. ArXiv (2020).

[31] Chang Wei Tan, Christoph Bergmeir, Francois Petitjean, and Geoffrey I Webb. 2020. Time Series Regression. arXiv preprint arXiv:2006.12672 (2020).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008.

[33] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. arXiv:2001.08317 [cs.LG]

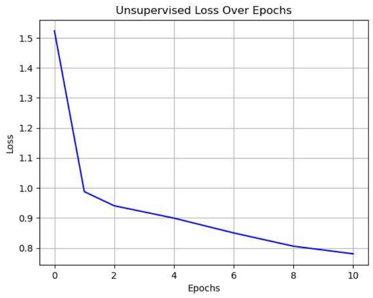
[34] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, C. Lumezanu, Wei Cheng, Jingchao Ni, B. Zong, H. Chen, and Nitesh V. Chawla. 2019. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In AAAI. https://doi.org/10.1609/aaai.v33i01.33011409

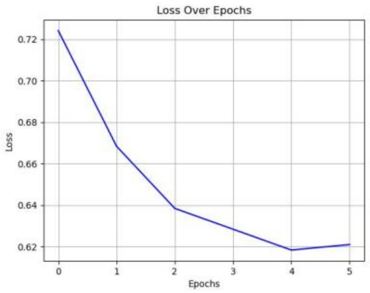
[35] Ruiqi Zhang, and Spencer Frei, 2023. Trained Transformers Learn Linear Models In-Context in arXiv:2306.09927v3 [stat.ML] 19 Oct 2023

[36] Ailing Zeng1, Muxi Chen, Lei Zhang and Qiang Xu, 2022. Are Transformers Effective for Time Series Forecasting? In arXiv:2205.13504v3 [cs.AI] 17 Aug 2022

A. ADDITIONAL MATERIAL

1. FACE DETECTION GRAPHS





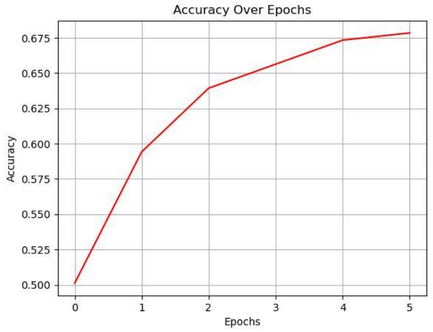
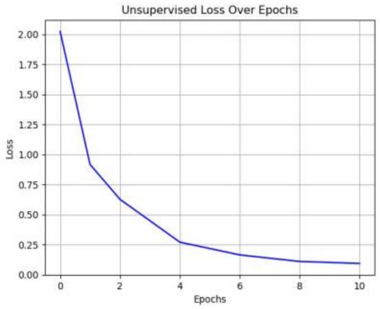
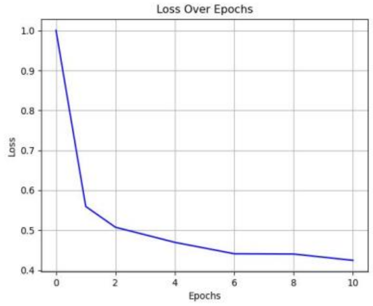


Figure 4: 1st Image: Unsupervised Loss, 2nd Image: Finetune Loss, 3rd Image: Finetune Accuracy

2. TISELAC GRAPHS:





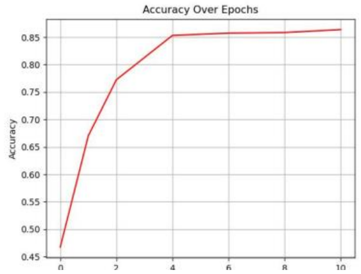


Figure 5:1st Image: Unsupervised Loss, 2nd Image: Finetune Loss, 3rd Image: Finetune Accuracy

3. JAPANESE VOWELS GRAPHS:

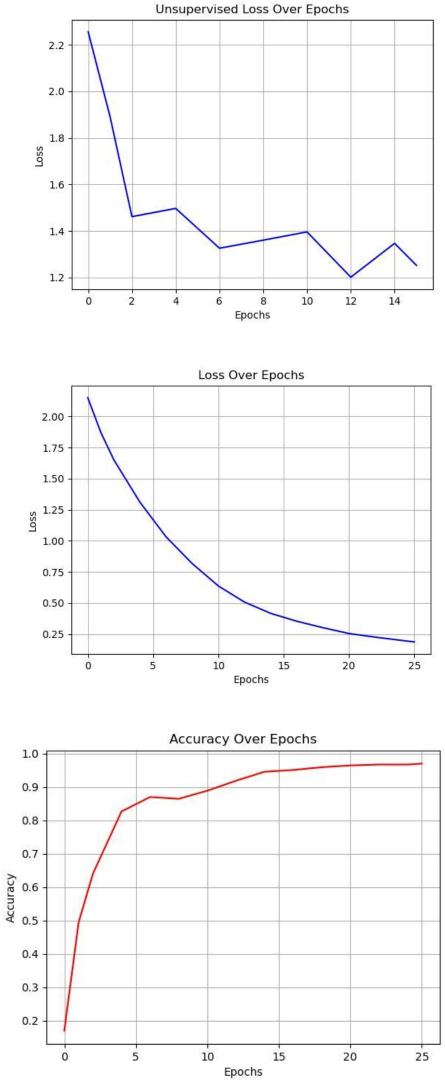


**Figure 6: 1st Image: Unsupervised Loss,
2nd Image: Finetune Loss,
3rd Image: Finetune Accuracy**

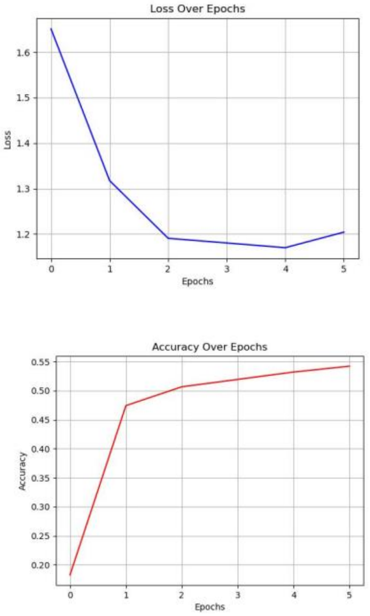
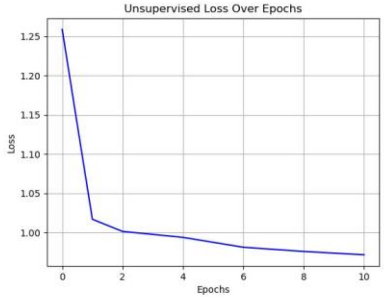


**Figure 7: 1st Image: Unsupervised Loss,
2nd Image: Finetune Loss,
3rd Image: Finetune Accuracy**

4. SLEEP GRAPHS:


Unsupervised Loss Over Epochs
Loss
Epochs