

# *Time2Vec and the Nuances of Temporal Representation*

*Efstathios Kotsis, Michael Darmanis  
Nektarios Christou, Vasileios Venieris*

*July 24, 2023*

Time2Vec, as described in the paper by Borealis et al.<sup>1</sup>, offers a vector representation of time in machine learning models. However, it can be susceptible to misinterpretation in various domains. This is an attempt to explore the potential pitfalls of Time2Vec, especially when juxtaposed against the transformer architecture.

<sup>1</sup> Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019

## *Introduction*

Time is a concept of an intricate and multifaceted nature. Within its vast expanse, one often encounters patterns. Some of these patterns are periodic, manifesting as predictable events like sales peaks during weekends or the rhythm of seasonal weather changes. On the other hand, there are non-periodic patterns, where certain events unfold unpredictably or become increasingly probable as time progresses. A prime example of this is how the risk of certain diseases can escalate with age.

Recognising these complexities inherent to time, Borealis et al.<sup>2</sup> created Time2Vec, a representation model specifically crafted to encapsulate the duality and uniqueness of time-patterns. At the heart of Time2Vec are three foundational principles, as laid out by its authors. First is the inherent duality of time-patterns, emphasising the coexistence of both predictable and unpredictable events. Second, there's the principle of invariance to time-scaling, asserting that a genuine representation of time should be consistent, whether one observes milliseconds or millennia. Lastly, the principle of model-compatibility stresses the importance of simplicity; the model should be able to integrate effortlessly with a wide array of other models, underscoring its broad relevance.

Given time's distinct nature, especially when juxtaposed against other features, the emergence of a specialized representation like Time2Vec becomes not just valuable, but essential. Time2Vec's mathematical framework represents time through the following formulation

$$t2v(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i, & \text{if } i = 0 \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{if } 1 \leq i \leq k, \end{cases}$$

where  $t2v(\tau)[i]$  denotes the representation of time  $\tau$  for the  $i$ -th com-

<sup>2</sup> Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019

ponent,  $\omega_i$  represents a frequency term for the  $i$ -th component,  $\varphi_i$  signifies a phase term for the  $i$ -th component and  $\mathcal{F}$  is a periodic function encapsulating repeating patterns in time.

In essence, this formula captures both linear and periodic patterns. The linear term addresses trends, while the periodic function captures recurring sequences in time, thereby producing a comprehensive time representation.

### *Incorporating Time2Vec in Predictive Modeling*

Integrating the Time2Vec technique can drastically optimize models, enabling them to discern and learn from the subtle temporal patterns and overarching trends inherent in data. Incorporating Time2Vec in a predictive model roughly looks like the following:

(add figure here)

The initial phase involves **Time Embedding**. The main objective of this stage is to transform raw temporal data elements, such as timestamps, days, and hours, into intricate embeddings. To achieve this, the Time2Vec module is employed, ensuring the embeddings encapsulate both linear and cyclical time-related patterns.

Following the embedding process is the **Model Integration** phase. Here, the aim is to cohesively integrate these time-based embeddings into predictive models. These embeddings are not isolated; they act as features and are integrated alongside other relevant data. Various models can harness these embeddings, from traditional neural networks to sophisticated architectures like recurrent neural networks (RNNs, including LSTM and GRU) and transformers.

Once integration is achieved, the process transitions to **Prediction**. At this juncture, the model's task is to generate predictions by leveraging the time embeddings and other incorporated features. Depending on the nature of the data and the intended outcome, this phase can be bifurcated. For continuous outcomes such as stock prices or temperatures, the model embarks on regression tasks. Conversely, when data needs to be categorized into specific classes, like discerning between "faulty" and "not faulty" states in machinery maintenance datasets, classification tasks are the focal point.

Post prediction, the process emphasizes **Training**. The cornerstone of this phase is to fine-tune the model. This is accomplished by juxtaposing the model's predictions against actual data values. The model's performance is quantified using specific loss functions. For regression-driven tasks, metrics like Mean Squared Error (MSE) or Mean Absolute Error (MAE) are apt, while classification-centric tasks predominantly rely on the Cross-Entropy Loss.

Lastly, **Optimization** is performed. Here, the overarching goal

is to augment both the model’s accuracy and its operational efficiency. This is carried out using optimization methods such as SGD or Adam. A crucial aspect of this phase is the dual optimization of both the primary model’s weights and the Time2Vec module’s parameters. The intention is to fine-tune the system to capture relevant temporal patterns with even greater precision.

### *Strengths*

1. Time2Vec’s ability to encapsulate the progression, periodicity, and scale of time is of great interest. The representation, while being versatile across datasets, captures time’s essence in a manner that’s more sophisticated than mere timestamps.
2. The transformer architecture, though originally crafted for NLP, has been employed in numerical timeseries classification, as seen in various articles (show articles). Time2Vec’s potential as a positional encoding mechanism in such architectures underscores its versatility.

**Rich Representation of Time:** Time2Vec’s main advantage lies in its ability to convert a simple timestamp into a multi-dimensional vector that captures both linear and periodic nuances of time. This is akin to how humans perceive time - not just as a point but in terms of day, week, month, and year.

**Versatility Across Datasets:** Instead of relying on handcrafted features that may vary between datasets, Time2Vec offers a systematic way to encode time, making it potentially adaptable across different time series datasets.

**Enhancement of Existing Models:** Time2Vec can be integrated with various machine learning models, including but not limited to transformers, LSTMs, and traditional regression models. By providing these models with a richer understanding of time, they can potentially make better predictions.

### *Concerns and Clarifications*

1. Many articles seem to use timeseries values (like prices or sales) as inputs to Time2Vec, rather than the actual temporal information (i.e., a time index  $t = 0, 1, 2, 3, \dots$ ). This approach seems more like an enrichment of feature representation rather than true positional encoding. Even the original Word2Vec paper<sup>3</sup>, which inspired Time2Vec, hints at its design being more aligned with time-related features.

<sup>3</sup> Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013

2. Terms like “positional encoding” have specific meanings. If Time2Vec is used to encode features rather than positions, it leads to potential confusion. Such confusion can compound when architectures are layered and interwoven in complex models.
3. Even if Time2Vec was applied on a time index, it should likely be a global time index created before batching or windowing the data. This is because the local time index within a batch might be arbitrary, potentially introducing inconsistencies from a theoretical point of view.
4. The use of sines in Time2Vec seems to be connected to a Fourier series. Indeed, Fourier series decomposes a function into sines and cosines, potentially capturing periodicities in the data. This raises a crucial question: Is Time2Vec’s mechanism analogous to Fourier transformation, and if so, how does it differ in its application?
5. The core idea behind Time2Vec is somewhat related to the Fourier Series in that both deal with the decomposition of signals (or time in the case of Time2Vec) into a set of frequencies. However, while the Fourier Series uses fixed sines and cosines as basis functions, Time2Vec allows these frequencies to be learned.
6. The paper references the use of up to 64 sinusoids. What does using multiple sinusoids imply? It is likely about transforming a single time point into multiple features;<sup>4</sup>
  - (a) **Increased Expressivity:** A single sinusoid can only capture one specific frequency or oscillation pattern. By using multiple sinusoids, each with potentially different frequencies and phases, the representation can capture a more complex and diverse set of patterns.
  - (b) **Time Decomposition:** Similar to the Fourier transform, which decomposes a signal into its constituent sinusoidal frequencies, using multiple sinusoids allows the representation to decompose the time information into various frequency components.
  - (c) **Higher Dimensional Embedding:** By transforming a single time point into multiple features (via multiple sinusoids), the time information is embedded into a higher-dimensional space. This can make it easier for machine learning models to discern patterns, as the higher-dimensional space may separate or disentangle features that are overlapped or conflated in the original time representation.
  - (d) **Flexibility:** The ability to learn the parameters of these sinusoids (amplitude, frequency, phase) means the model can adapt the representation to best fit the patterns in the data. For instance, certain frequencies might be more important for one

<sup>4</sup> Akin to how one perceives ‘now’ in terms of day, week, month, and year, such a multi-sinusoidal approach might be an automated way to capture various cyclical patterns in time data.

dataset, while others might be more important for a different dataset.

- (e) **Interactions with Other Features:** When a single time point is transformed into multiple features, it provides more avenues for interactions with other features in the data.

7. It employs both linear and sinusoidal transformations. Which features should undergo a linear transformation? Without the original code from the paper, this remains an area of ambiguity and warrants further exploration.

## *Conclusion*

Time2Vec, in its essence, presents an alternative approach to representing time. By using multiple sinusoids to transform a single time point, it provides a richer, more expressive representation of time, which can capture a diverse range of patterns and make it easier for models to learn from time-based data. However, its application needs careful consideration. Misinterpretations, especially in areas as intricate as positional encoding, can lead to flawed models and misguided conclusions. A more rigorous mathematical framework and clearer guidelines for Time2Vec’s application are required.

## *Acknowledgements*

This report was typeset using L<sup>A</sup>T<sub>E</sub>X, originally developed by Leslie Lamport and based on Donald Knuth’s T<sub>E</sub>X. A template that can be used to format documents with this look and feel has been released under the permissive APACHE LICENSE 2.0, and can be found online at <https://tufte-latex.github.io/tufte-latex/>.

## *References*

- Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.