Enhancing Remaining Time Prediction in Business Process Monitoring via Cross-Entropy Supervised Entity Embeddings and Transformer Model

MAMERTHE WABIWA MUBAKE[1,2], WITESYAVWIRWA VIANNEY KAMBALE[3], KYANDOGHERE KYAMAKYA[1]

[1]Institute for Smart Systems Technologies,
Universität Klagenfurt,
AUSTRIA

[2]Faculty of Science and Technology,
Université Libre des Pays des Grands Lacs, Goma,
DEMOCRATIC REPUBLIC OF THE CONGO

[3]Faculty of Information and Communication Technology,
Tshwane University of Technology, Pretoria,
SOUTH AFRICA

Abstract: Accurate prediction of the remaining time for ongoing business process instances is crucial for making proactive decisions, meeting deadlines, and optimizing resource allocation. This paper introduces a Transformer-based multitask learning framework that improves time prediction by incorporating an auxiliary classification task. The auxiliary task supervises the learning of entity embeddings for categorical attributes in event logs, using cross-entropy loss to guide the model toward more meaningful representations. By combining temporal modeling with supervised embedding learning, the architecture addresses two core challenges in process data: capturing sequence dependencies and understanding categorical relationships. Experiments on a real-world container terminal data set show that the proposed approach reduces the absolute mean error by 48.4% and the square root error by 39.8% compared to established baselines. These results demonstrate that embedding supervision significantly improves predictive performance and can improve the reliability of time-related forecasts in business process monitoring.

Key-Words: Business Process Monitoring, Predictive Process Monitoring, Transformer, Entity Embeddings, Multitask Learning, Remaining time prediction

Received: March 13, 2025. Revised: December 7, 2025. Accepted: December 16, 2025. Published: December 31, 2025.

1 Introduction

Predictive Business Process Monitoring (PBPM) is an essential research direction in Business Process Management (BPM) that aims to forecast the future behavior of ongoing process instances using historical execution data stored in event logs, [1], [2]. By offering proactive insights into running processes, PBPM helps organizations improve resource utilization, reduce the risk of deadline violations, and take timely corrective actions to ensure better outcomes, [3], [4]. Nevertheless, LSTM-based models face difficulties preserving long-term dependencies because of their fundamentally sequential architecture, [5].

Furthermore, numerous methodologies depend on simplistic encoding techniques for categorical data, which do not adequately represent their underlying semantic relationships, [6]. Transformer-based designs, [7], have emerged as a viable solution to address these restrictions. Their self-attention mechanism facilitates more efficient modeling of long-range interdependence and global context, rendering them especially appropriate for business process data characterized by complex temporal patterns, [8]. Simultaneously, entity embeddings have demonstrated considerable efficacy in enhancing the representation of categorical variables in tabular data, [9].

Nonetheless, a significant difficulty persists due to the absence of explicit supervision during learning these embeddings, potentially impeding their capacity to represent genuine semantic links accurately. Recently, academics have commenced the application of Transformer models explicitly to PBPM problems. The study, [10], introduced the ProcessTransformer, which employs self-attention to acquire high-level representations of event sequences, showing enhancements in forecasting subsequent activities and remaining time. Similarly, [11], presented the HiP-Transformer, a hierarchical framework that partitions process traces using idea drift detection and identifies dependencies across various levels of granularity. These methodologies underscore the potential of Transformers in this field, although they mostly neglect the significance of embedding supervision in enhancing categorical feature representation.

This paper proposes a Transformer-based multi-task learning framework designed to enhance the prediction of remaining time in business processes. Our method introduces an auxiliary classification objective to guide the learning of entity embeddings for categorical attributes within event logs. Unlike conventional multi-task learning, which targets multiple prediction outputs, our auxiliary task serves as a regularization mechanism to structure the embedding space and better capture semantic relationships among categorical features. By combining this embedding supervision with a regression head for time prediction, the model effectively learns both temporal dependencies and rich categorical representations.

The remainder of this paper is organized as follows: Section 2 reviews related work in predictive business process monitoring and embedding techniques. Section 3 presents the methodology of the proposed approach, detailing the model architecture and the auxiliary classification task. Section 4 describes the experimental setup, including datasets, evaluation metrics, and baseline models. Section 5 concludes the paper and outlines directions for future research.

2 Related Works

Techniques for predictive business process monitoring focus on predicting the future behavior of ongoing process executions by leveraging models learned from historical event data. This section provides an overview of relevant methodologies and then identifies a specific research gap that forms the foundation of our study. Existing approaches to predictive process monitoring include several prediction tasks, among them those focused on predicting outcomes and those related to time prediction.

2.1 Prediction of Case Outcome

Business Process Management (BPM) involves systematically managing work to achieve predictable results and optimize processes, [12]. Process mining, as a data-driven technology for the analysis of business processes, has evolved from its initial focus on discovering process models from historical data to include predictive capabilities, [2]. This shift from descriptive to predictive analytics has led to the emergence of predictive business process monitoring (PBPM) as a novel research stream within the BPM domain, [13].

The study, [14], tackle the challenge of predicting the most probable outcome of a given process instance. Prior methods have predominantly relied on straightforward symbolic sequence classification, where event trace features are extracted as sequences of labels to train classifiers for runtime predictions. However, this approach may be limited in capturing complex process dynamics.

The study, [15], introduced a predictive monitoring framework aimed at estimating the likelihood that specific conditions, or predicates, will be satisfied during the execution of a process instance. Their method integrates both the sequence of events and their corresponding data attributes. The approach unfolds in two stages: initially, it clusters completed case prefixes based on control flow characteristics; subsequently, it trains a classifier for each cluster using attribute information to differentiate between predicate satisfaction and violation.

In [4], the authors proposed a prediction method built on the attention mechanism, originally developed for natural language processing and neural machine translation. Their model uses the full set of hidden states to anticipate upcoming activities and overall process outcomes, highlighting the attention mechanism's capacity to recognize relevant patterns in process behavior.

The study, [8], devised a novel framework that integrates multi-view learning with deep learning methodologies to improve predictive performance in PBPM. Their technique encompasses several dimensions of event logs by simultaneously analyzing multiple data views. Experimental findings across benchmark datasets demonstrated the model's superiority over contemporary state-of-the-art methodologies, highlighting the advantages of a more cohesive perspective on process execution.

The study, [5], introduced a multi-task learning approach that integrates BERT with transfer learning for business process monitoring. Their approach presents the Masked Activity Model (MAM) as a pre-training job to cultivate generic process trace embeddings. The authors illustrated that this pre-training, succeeded by fine-tuning, enhances effective knowledge transfer across diverse PBPM tasks, rendering model training more efficient and flexible to varying prediction objectives.

The study, [16], expanded the application of LSTM models to outcome prediction in PBPM, beyond conventional next-activity forecasting. Their research demonstrated how deep learning architectures may tackle more extensive prediction challenges by incorporating the broader context of full process instances, highlighting the adaptability of neural models in process mining applications.

2.2 Prediction of Time-Related Properties

Forecasting time-related attributes, especially the remaining duration of active process instances, is essential for companies to enhance resource allocation, prevent deadline infringements, and deliver precise client estimations, [17]. This predictive ability facilitates proactive intervention and enhanced planning, making it crucial to monitor business processes.

In [18], the authors examined Long Short-Term Memory (LSTM) neural networks to create precise models for diverse predictive process monitoring tasks. LSTMs demonstrated superior performance compared to existing approaches in forecasting the subsequent event and its timestamp. They additionally employed these models to predict the complete progression of an ongoing case. They utilized the identical methodology to forecast remaining time, attaining superior outcomes compared to conventional, task-specific methods. This study illustrated the adaptability of LSTM networks across various predictive tasks in PBPM.

The study, [19], enhanced LSTM-based methodologies to forecast the remaining time. Their data-aware LSTM networks integrated the control flow and data perspectives of process executions, showing substantial enhancements compared to prior methods. Their research emphasized the significance of analyzing the sequence of operations and the relevant data properties for precise time estimations.

In [20], the authors compared Classification-Based and Regression-Based Predictive Process Monitoring models, emphasizing remaining time prediction across various processes. Essential evaluation criteria encompassed accuracy, training duration, and real-time prediction velocity, emphasizing the influence of model configurations on performance. Their thorough comparison offers significant insights into the advantages and drawbacks of various modeling techniques for time prediction tasks.

The study, [21], performed a systematic evaluation and cross-benchmark analysis of the existing time prediction methods. Their research assessed diverse methodologies across numerous real-world datasets, thoroughly evaluating the current advancements in remaining time prediction. Their findings provide crucial criteria for choosing suitable predictive models based on process attributes and forecasting needs.

Despite the encouraging outcomes of LSTMs, they encounter constraints in identifying long-term dependencies owing to their sequential processing characteristics. The advent of Transformer designs, [7], has provided novel opportunities for modeling process data, as they more effectively capture long-range interdependence via self-attention processes.

The study, [10], introduced *ProcessTransformer*, a Transformer-based architecture to acquire high-level representations from business process event logs. Their methodology employed self-attention to explicitly capture global dependencies inside event sequences, producing promising outcomes across various PBPM tasks, such as next activity prediction, event time prediction, and remaining time prediction. Their methodology revealed that attention-based processes can function as viable substitutes for recurrent models in capturing temporal patterns in event data.

Expanding on this approach, [11], presented the HiP-Transformer, a hierarchical model that divides business process traces into subsequences by idea drift detection. The approach then employs attention-based encoding at three tiers: event, subsequence, and case, to capture structural and temporal dependencies across varying granularities. Their framework shows enhancements in remaining time and the precision of subsequent activity predictions relative to conventional transformer-based and LSTM approaches, notably through superior management of variability and fluctuations in business process behavior.

The study, [6], showed that Entity Embedding can improve the precision of Deep Neural Networks (DNNs) with tabular data containing categorical variables. To assess the method's resilience, scientists performed experiments utilizing synthetic and real-world datasets, juxtaposing its performance against other supervised learning regression techniques. Their methodology tackles a significant issue in process mining: the appropriate representation of standard categorical variables in event logs.

Notwithstanding these breakthroughs, most methodologies employ rudimentary encoding techniques for categorical features, which inadequately represent intricate semantic associations, [9]. Entity embeddings provide a refined representation of categorical traits; yet, ensuring that these embeddings effectively reflect semantic relationships is difficult. This is particularly important for process mining, where the meaning and relationships between categorical attributes (such as activity names, resources, or departments) are crucial for accurate predictions.

This paper adopts the latter approach as a baseline and demonstrates how combining Entity Embedding with a Transformer-based architecture and cross-entropy loss can further enhance the accuracy of remaining time prediction, while offering improved generalizability. The proposed approach specifically addresses the challenge of learning meaningful representations for categorical attributes through a novel learning framework that introduces an auxiliary classification task to guide the learning of entity embeddings.

3 Methodology

The proposed Transformer-based model aims to improve remaining time prediction by incorporating an auxiliary classification task that supervises the learning of entity embeddings for categorical attributes found in event logs, as shown in Figure 1

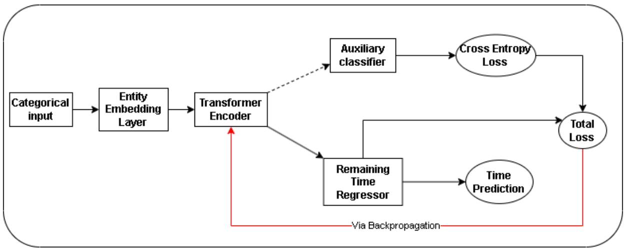

Fig. 1: Transformer-based model with entity embeddings guided by an auxiliary classification loss (cross-entropy) to improve representation learning for remaining time prediction. *Source: created by the authors.*

3.1 Preliminaries

3.1.1 Event Log

Process mining requires event logs as input. In practice, event logs can vary significantly. However, all event logs share a common characteristic: they display occurrences of events at specific moments in time, where each event refers to a particular process and an instance thereof, i.e., a case, [2]. Table 1 is an event log fragment that illustrates the typical information in an event log. In process mining, any event can be related to both a case and an activity, and the events within a case are ordered. Therefore, the "case id" and "activity" columns in Table 1 represent the bare minimum for process mining, [1].

Table 1: A fragment of event logs from the Container Terminal dataset; *Source: created by the authors.*

Case id	Activity	Start Timestamp	End Timestamp
AAAU9001220-2021-06-28	BAPLIE	27/06/2021 10:50	27/06/2021 11:09
AAAU9001220-2021-06-28	VESSEL_ATB	28/06/2021 18:35	28/06/2021 20:00
AAAU9001220-2021-06-28	DISCHARGE	28/06/2021 20:51	28/06/2021 22:22
AAAU9001220-2021-06-28	STACK	28/06/2021 22:19	28/06/2021 22:40
AAAU9001220-2021-06-28	HAS_QUARANTINE_FLAG	02/07/2021 07:45	02/07/2021 09:06
AAAU9001220-2021-06-28	CUSTOMS_DEL	04/07/2021 22:49	05/07/2021 00:00

As shown in Table 1, the example case shows a sequence of activities for a container, starting with BAPLIE (container loading plan), followed by vessel arrival (VESSEL_ATB), unloading (DISCHARGE), placement in the yard (STACK), quarantine flagging, and customs processing. This represents the container terminal process we're analyzing.

3.1.2 One-Hot Encoding

In machine learning applications, particularly those involving neural networks, categorical variables pose unique challenges. One common approach is to encode categorical variables into integer labels. However, this method is problematic as it introduces an artificial ordinal relationship, the model will interpret higher label numbers as having greater importance than lower ones, [6]. Therefore, an encoding method that assigns equal weight to all categorical values is necessary.

One-hot encoding addresses this issue by representing each categorical value as a binary vector, where only one element has a value of 1, and all others have a value of 0, [22]. These resulting variables are also referred to as dummy variables. As illustrated in Figure 2, categorical values (e.g., "cat", "dog", "bird") are transformed into binary vectors where only the position corresponding to that specific category contains a 1.

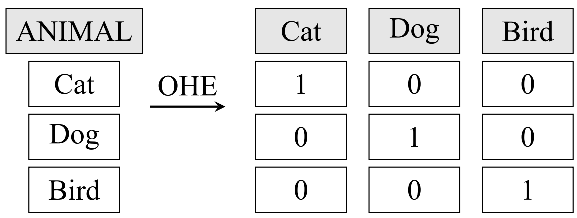

Fig. 2: Illustration of One-Hot Encoding(OHE); *Source: created by the authors.*

While one-hot encoding solves the ordinal relationship problem and works well with neural network architectures, it has significant disadvantages when applied to process data with high-cardinality categorical variables. The number of binary columns produced can become extremely large, resulting in sparse high-dimensional input vectors that lead to inefficient learning and poor generalization, [9]. Moreover, semantically similar values (such as related activities in a process) are not placed close together in the encoding space, meaning that the intrinsic relationships between features are not captured, [23]. This limitation is particularly problematic in process mining, where understanding the relationships between activities is crucial for accurate prediction and analysis.

3.1.3 Entity Embedding

Entity embedding is a technique that can overcome the limitations of one-hot encoding by mapping categorical variables to dense vectors in a multi-dimensional space, [9]. Similar to word embeddings used in Natural Language Processing, entity embeddings represent categorical values as points in a continuous space where semantically identical values are positioned closer together.

The key advantage of entity embedding is that it can capture the intrinsic properties and relationships between categorical values. As shown in Figure 3, different categorical values are mapped to different points in a continuous embedding space, allowing the model to learn meaningful representations that reflect the semantic similarities between categories.

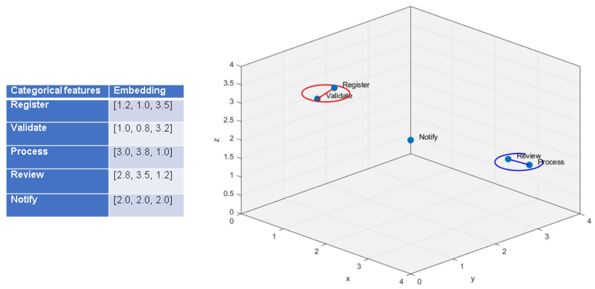

Fig. 3: Visual representation of Entity Embedding where each process activity is projected into a continuous multidimensional space. Similar activities, such as 'Register' and 'Validate', are positioned closer to each other, as indicated by the red circle. Similarly, 'Process' and 'Review' activities are grouped together (blue circle) based on their semantic similarity. *Source: created by the authors.*

Entity embeddings are typically learned jointly with the main task through backpropagation. However, this approach can be limited by the absence of explicit supervision for the embedding learning process, which may result in suboptimal representations that do not fully capture the complex relationships between categorical values, [24].

3.1.4 Transformer Architecture

The Transformer is a neural network architecture presented by [7], that has transformed sequence modeling jobs. In contrast to recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), which handle data sequentially, Transformers process complete sequences concurrently, enhancing training efficiency.

The primary novelty of Transformers is the self-attention mechanism, enabling the model to assess the significance of various points in the input sequence during prediction. This allows Transformers to more efficiently capture long-range dependencies compared to RNNs, which sometimes have difficulties with information from remote points in the sequence, [25].

The original Transformer architecture has an encoder and a decoder; however, numerous applications, like the method suggested in this study, employ solely the encoder component.

3.2 Proposed Approach

The proposed approach uses a Transformer-based architecture to process sequences of categorical inputs, which are first transformed through an entity embedding layer. To improve the quality of these embeddings, an auxiliary classification objective is introduced during training. This auxiliary task, optimized using cross-entropy loss, serves only as a regularization signal to guide the embedding space toward more informative representations. The primary task is predicting the remaining time, which is handled through a regression head. The total loss, which combines the regression loss with the auxiliary cross-entropy component, is backpropagated jointly, as illustrated in Figure 1; however, only the regression output is used during inference.

3.2.1 Problem Formulation

Predictive business process monitoring aims to forecast the future behavior of ongoing process executions by leveraging models learned from historical event data, [13]. This study focuses specifically on the task of predicting remaining time.

Given a prefix of a case (i.e., a sequence of activities and associated attributes up to a certain point), our goal is to predict the remaining time until the case completes. Formally, we define the remaining time prediction problem as follows:

Let $\sigma = \langle e_1, e_2, ..., e_n \rangle$ be a complete trace of a case, where each event e_i includes an activity name and other attributes. Given a prefix $\sigma_{prefix} = \langle e_1, e_2, ..., e_k \rangle$ where $k < n$, the task is to predict the remaining time $RT(e_k) = timestamp(e_n) - timestamp(e_k)$, which is the time between the last event in the prefix and the last event in the complete trace.

3.2.2 Entity Embedding Layer

The entity embedding layer transforms categorical inputs into dense vector representations. For each categorical attribute in the event log (such as activity names, resources, departments), we create a separate embedding matrix:

$$E_{cat_i} \in \mathbb{R}^{n_i \times d} \tag{1}$$

where n_i is the number of unique values for the categorical attribute i and d is the embedding dimension.

The embedding dimension is determined based on the cardinality of the corresponding categorical variable, following the heuristic proposed by [9]:

$$d = \min\left(50, \left\lfloor \frac{n_i + 1}{2} \right\rfloor\right) \quad (2)$$

This adaptive sizing ensures that higher-cardinality variables have more capacity to represent their semantic space while avoiding unnecessarily large embeddings for low-cardinality variables.

3.2.3 Transformer Encoder

The Transformer encoder processes the embedded sequence to capture temporal dependencies and relationships between events. Our implementation follows the architecture proposed by [7], with modifications tailored to process mining data.

The key components of our Transformer encoder include:

Multi-head Self-attention : The self-attention mechanism computes attention scores between all pairs of positions in the sequence:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where Q (queries), K (keys), and V (values) are linear projections of the input embeddings, and d_k is the dimension of the key vectors.

Multi-head attention allows the model to jointly attend to information from different representation subspaces:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O \quad (4)$$

where each $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and the W matrices are learnable parameters.

Position-wise Feed-forward Networks : After the attention mechanism, each position in the sequence is processed by a feed-forward network:

$$F(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (5)$$

This allows the model to transform the attention outputs and introduce non-linearity.

Layer Normalization and Residual Connections : To stabilize and accelerate training, we apply layer normalization and residual connections:

$$\text{LayerNorm}(x + \text{Sublayer}(x)) \quad (6)$$

where $Sublayer(x)$ represents either the multi-head attention or the feed-forward network.

The final output of the Transformer encoder provides contextualized representations for each position in the sequence, capturing both the sequential patterns and the interactions between different categorical attributes.

3.2.4 Auxiliary Classifier and Cross-entropy Loss

The key innovation of the proposed approach is the introduction of an auxiliary classification task specifically designed to guide the learning of entity embeddings. This auxiliary task predicts the next activity in the process based on the current prefix.

The auxiliary classifier comprises a softmax layer that receives the contextualized representation from the Transformer encoder and produces a probability distribution over potential subsequent activities:

$$p(a_{next}|\sigma_{prefix}) = \text{softmax}(W_c h + b_c) \quad (7)$$

where h is the hidden representation from the Transformer encoder, and W_c and b_c are learnable parameters.

The classifier is trained using cross-entropy loss:

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(p_i) \quad (8)$$

where y_i is the genuine label (one-hot encoded) and p_i denotes the expected probability for class i.

This auxiliary job explicitly supervises the embedding layer, directing it to acquire representations that encapsulate the process flow and the interrelations among activities. The model learns to encode information regarding transition probabilities and sequential patterns by forecasting the subsequent activity, which is essential for estimating remaining time.

3.2.5 Remaining Time Regressor

The primary objective of the presented model is to forecast the time remaining till case completion. This is executed via a regression head that utilizes the contextualized representation from the Transformer encoder to generate a continuous value:

$$\hat{y}_{RT} = \text{ReLU}(W_r h + b_r) \quad (9)$$

where W_r and b_r are learnable parameters, and the ReLU activation ensures non-negative predictions.

The regression task is trained using mean squared error (MSE) loss:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (10)$$

where y_i is the true remaining time and $\hat{y}_i$ is the predicted remaining time for instance i.

3.2.6 Joint Optimization

The overall loss is calculated as a weighted aggregate of the regression and classification losses:

$$L_{total} = L_{MSE} + \alpha \cdot L_{CE} \quad (11)$$

where α serves as a hyperparameter that equilibrates the influence of the two loss components.

The suggested approach enhances representations of categorical features by simultaneously optimizing both objectives. The cross-entropy loss directs the embedding space to encapsulate pertinent semantic links associated with the process. Conversely, the regression loss guarantees that these associations facilitate the prediction of the remaining time.

This learning methodology mitigates a significant drawback of conventional entity embedding techniques: the absence of explicit guidance in acquiring semantic links among categorical values. We implement a supplementary classification task utilizing cross-entropy loss to furnish direct supervision for the embedding layer. This yields more significant representations for categorical variables and, thus, more precise estimations of remaining time.

4 Experiments and Results

Extensive experiments were performed utilizing a real-world event log dataset to assess the efficacy of the suggested methodology. This section delineates the experimental configuration, encompassing the dataset attributes, preparation methodologies, and implementation specifics. The experimental findings are subsequently given and analyzed, contrasting the performance of the MultiTask Transformer model with several baseline methods. The assessment concentrates on two primary elements: the predictive accuracy of remaining time forecasts, quantified via conventional error measures, and the efficacy of the auxiliary classification job in facilitating entity embedding learning. These experiments aim to illustrate that integrating the Transformer architecture with cross-entropy directed entity embeddings yields substantial enhancements compared to conventional techniques.

The experiments were conducted using the "Event Log Dwelling Time Dataset", [26], a real-world dataset from a Container Terminal that tracks the time from when containers are unloaded from ships until they leave the port terminal. This dataset was collected over a 3-month period in 2021 and contains 952,069 events with multiple categorical attributes (container type, container size, yard block, yard slot, and document type) along with temporal information.

Following established practices in predictive business process monitoring [18], several preprocessing steps were performed. First, 54 rows with missing timestamps (0.006% of the dataset) were removed. Then, for each event, the remaining time (in minutes) until case completion was calculated by subtracting the current event's timestamp from the case completion timestamp. Significant variance in the remaining time distribution was observed (mean: 3,208 minutes, std: 88,607 minutes), with extreme outliers reaching up to 35,250,080 minutes. To address this, values were clipped to the 0.1 and 99.9 percentiles, resulting in a more balanced distribution (mean: 2,964 minutes, std: 4,282 minutes). For sequence modeling, prefixes with lengths ranging from 1 to 10 events were generated, resulting in 892,966 prefixes. To augment the training data, variations with small amounts of random noise were added to the remaining time values, resulting in a total of 1,594,636 instances. Categorical features were encoded using both traditional approaches (one-hot encoding) and entity embeddings to allow for fair comparison between methods.

To evaluate the predictive performance of models, two widely used metrics for regression tasks were employed: Mean Absolute Error (MAE), which measures the average absolute difference between predicted and actual remaining time values in minutes, and Root Mean Square Error (RMSE), which calculates the square root of the average squared differences between predicted and actual values, giving higher weight to larger errors. For the auxiliary classification task in the proposed multi-task model, classification accuracy was additionally measured, representing the proportion of correctly predicted next activities.

All models were implemented using PyTorch and trained on an NVIDIA GeForce GTX 1650 GPU with 4GB of memory. The Adam optimizer was employed with a learning rate of 0.001. Early stopping was used to prevent overfitting, with patience set to 10 epochs for baseline models and 20 epochs for the transformer model.

Several baseline models were implemented to evaluate the effectiveness of the proposed approach, including a basic deep neural network that uses encoded categorical features directly and a deep neural network that uses one-hot encoded categorical features. This deep neural network utilizes entity embeddings for categorical features, but without the transformer architecture or multi-task learning components, and an extended version of the entity embedding model that incorporates additional process-related features.

Figure 4 shows the training and validation loss curves for our MultiTask Transformer model. The model demonstrated stable convergence, with both total loss and component losses (regression and classification) showing consistent improvement over the training epochs. Early stopping was triggered after 62 epochs, indicating that the model had reached optimal performance.

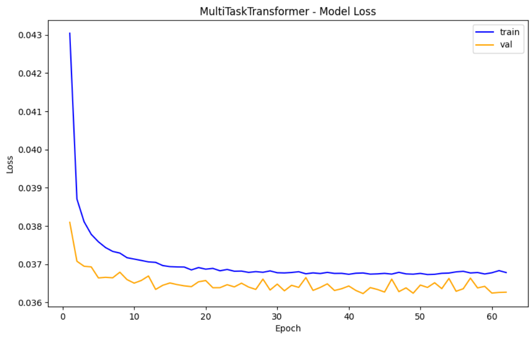

Fig. 4: Training and validation loss curves for the MultiTask Transformer model. *Source: created by the authors.*

Table 2 presents the performance of all models on the test set in terms of MAE and RMSE for remaining time prediction.

Table 2: Experimental Results for the Dwelling Time Process Event Log in RMSE (minutes) and MAE (minutes); *Source: created by the authors.*

Model	MAE	RMSE
Basic DNN	2,860.74	4,766.08
DNN + One-Hot Encoding	2,779.06	4,926.84
DNN + Entity Embedding	2,756.31	4,925.21
Enhanced DNN + Entity Embedding	2,742.87	4,909.75
MultiTask Transformer	**1,414.49**	**2,954.75**

The proposed MultiTask Transformer model achieved significantly better performance than all baseline models, with an MAE of 1,414.49 minutes and RMSE of 2,954.75 minutes. This represents a 48.4% reduction in MAE and 39.8% reduction in RMSE compared to the best baseline model (Enhanced DNN + Entity Embedding). Additionally, the MultiTask Transformer achieved 93.75% accuracy in predicting the next activity, validating the effectiveness of the auxiliary classification task. This high classification performance suggests that the model has successfully learned meaningful representations of process activities, which in turn contributes to improved remaining time predictions.

The experimental results confirm several key findings. The consistent improvement observed from basic encoding to one-hot encoding to entity embedding validates the hypothesis that better representations of categorical attributes lead to more accurate predictions. The substantial performance gap between the best baseline model and the MultiTask Transformer demonstrates that the transformer architecture effectively captures the temporal dependencies and complex patterns in process data. The auxiliary classification task offers significant guidance for the embedding layer, yielding more substantive representations that improve the efficacy of the primary regression job.

Moreover, these findings are consistent with current Transformer-based methodologies in PBPM. The ProcessTransformer model, presented by [10], revealed that self-attention mechanisms effectively capture dependencies in process data. Conversely, the HiP-Transformer system introduced by [11], utilized hierarchical encoding to enhance predictive accuracy. In contrast, our solution incorporates a multi-task framework with directed entity embedding learning, offering an alternate strategy for improving performance in time prediction. Transformer-based PBPM models demonstrate observed enhancements and validate that attention-based architectures can provide substantial advantages in intricate prediction contexts when augmented by supervisory signals.

These results underscore that incorporating Transformer-based architectures with supplementary learning objectives presents a promising avenue for predictive business process monitoring, especially with remaining time prediction tasks.

5 Conclusion and Future Work

This study, based on mathematical equations (1)–(11), introduced a novel method to predict the remaining time of ongoing business process instances, combining the strengths of Transformer models with entity embeddings enhanced through a supervised learning signal. At the core of this approach lies an auxiliary classification task, designed to guide the model in learning richer representations of categorical attributes, which represent a common yet challenging aspect of process data.

Empirical findings from a real-world container terminal dataset validated the efficacy of this architecture. The proposed strategy attained a

48.4% decrease in mean absolute error (MAE) and a 39.8% decrease in root mean square error (RMSE) relative to the most robust baseline. Furthermore, the classification task achieved an accuracy rate of 93. 75%, further strengthening its contribution to improving the overall learning process.

Three elements underpin these improvements: the semantic capacity of entity embeddings, the ability of Transformers to model long-range dependencies in sequences, and the added supervision from the auxiliary task that shapes the embeddings during training. Together, these components form a cohesive framework that better captures the structure and dynamics of business processes.

Beyond its predictive performance, the proposed approach highlights a general strategy for embedding supervision that could benefit other machine learning tasks involving structured categorical data.

Future work will focus on outcome prediction and resource allocation. We also aim to investigate alternative auxiliary tasks to further enhance embedding quality and assess the interpretability of the learned embeddings.

While this study focused on improving embedding quality through guided supervision, we did not analyze the impact of embedding dimensionality. A detailed sensitivity analysis, particularly with and without guidance, remains an open area for further investigation.

Finally, we plan to address concept drift to ensure the model remains effective in dynamic business environments.

References

[1] Marlon Dumas, L Marcello Rosa, Jan Mendling, and A Hajo Reijers, *Fundamentals of business process management*. Springer, 2018. DOI: 10.1007/978-3-662-56509-4.

[2] Wil Van Der Aalst, "Data science in action," in *Process mining: Data science in action*, Springer, 2016, pp. 3–23. DOI: 10.1007/978-3-662-49851-4_1.

[3] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan, "Business process intelligence," *Computers in industry*, vol. 53, no. 3, pp. 321–343, 2004. DOI: 10.1016/j.compind.2003.10.007.

[4] Abdulrahman Jalayer, Mohsen Kahani, Amin Beheshti, Asef Pourmasoumi, and Hamid Reza Motahari-Nezhad, "Attention mechanism in predictive business process monitoring," in *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, IEEE, 2020, pp. 181–186. DOI: 10.1109/EDOC49727.2020.00030.

[5] Hang Chen, Xianwen Fang, and Huan Fang, "Multi-task prediction method of business process based on bert and transfer learning," *Knowledge-Based Systems*, vol. 254, p. 109 603, 2022. DOI: 10.1016/j.knosys.2022.109603.

[6] N. A. Wahid, T. N. Adi, H. Bae, and Y. Choi, "Predictive business process monitoring – remaining time prediction using deep neural network with entity embedding," *Procedia Computer Science*, vol. 161, pp. 1080–1088, 2019. DOI: 10.1016/j.procs.2019.11.219.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008. DOI: 10.48550/arXiv.1706.03762.

[8] V. Pasquadibisceglie, A. Appice, G. Castellano, and D. Malerba, "A multi-view deep learning approach for predictive business process monitoring," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2382–2395, 2022. DOI: 10.1109/TSC.2021.3051771.

[9] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," *arXiv preprint arXiv:1604.06737*, 2016. DOI: 10.48550/arXiv.1604.06737.

[10] Zaharah A. Bukhsh, Aaqib Saeed, and Remco M. Dijkman, *Processtransformer: Predictive business process monitoring with transformer network*, 2021. DOI: 10.48550/arXiv.2104.00721. arXiv: 2104.00721 [cs.LG].

[11] Weijian Ni, Gang Zhao, Tong Liu, Qingtian Zeng, and Xingzong Xu, "Predictive business process monitoring approach based on hierarchical transformer," *Electronics*, vol. 12, no. 6, 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12061273.

[12] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo, "Automated discovery of process models from event logs: Review and benchmark," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 4, pp. 686–705, 2018. DOI: 10.1109/TKDE.2018.2841877.

[13] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini, "Predictive monitoring of business processes," in *Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings 26*, Springer, 2014, pp. 457–472. DOI: 10.1007/978-3-319-07881-6_31.

[14] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, and F. M. Maggi, "Complex symbolic sequence encodings for predictive monitoring of business processes," in *Business Process Management: 13th International Conference, BPM 2015*, 2015, pp. 297–313. DOI: 10.1007/978-3-319-23063-4_21.

[15] C. D. Francescomarino, M. Dumas, F. M. Maggi, and I. Teinemaa, "Clustering-based predictive process monitoring," *IEEE Transactions on Services Computing*, vol. 12, no. 6, pp. 896–909, 2019. DOI: 10.1109/TSC.2016.2645153.

[16] I. Teinemaa, M. Dumas, M. La Rosa, and F. M. Maggi, "Outcome-oriented predictive process monitoring: Review and benchmark," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 2, pp. 1–57, 2019. DOI: 10.1145/3301300.

[17] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni, "Time and activity sequence prediction of business process instances," *Computing*, vol. 100, no. 9, pp. 1005–1031, 2018. DOI: 10.1007/s00607-018-0593-x.

[18] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas, "Predictive business process monitoring with lstm neural networks," in *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29*, Springer, 2017, pp. 477–492. DOI: 10.1007/978-3-319-59536-8_30.

[19] Nicolo Navarin, Beatrice Vincenzi, Mirko Polato, and Alessandro Sperduti, "Lstm networks for data-aware remaining time prediction of business process instances," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2017, pp. 1–7. DOI: 10.1109/SSCI.2017.8285184.

[20] R. Aalikhani, M. Fathian, and M. Reza Rasouli, "Comparative analysis of classification-based and regression-based predictive process monitoring models for accurate and time-efficient remaining time prediction," *IEEE Access*, vol. 12, pp. 67 063–67 093, 2024. DOI: 10.1109/ACCESS.2024.3397185.

[21] Ilya Verenich, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, and Irene Teinemaa, "Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 4, pp. 1–34, 2019. DOI: 10.1145/3331449.

[22] Sebastian Raschka and Vahid Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd, 2019, ISBN: 978-1-78995-575-0.

[23] H. Zhang, S. Zheng, and J. Ye, "A decomposable attention model for natural language inference," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. DOI: 10.48550/arXiv.1606.01933.

[24] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016. DOI: 10.1145/2959100.2959190.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019. DOI: 10.18653/v1/N19-1423.

[26] H. N. Prasetyo, R. Sarno, K. R. Sungkono, I. Waspada, and R. Budiraharjo, *Event log dwelling time dataset*, Mendeley Data, V2, 2024. DOI: 10.17632/yvp2b4rtp3.2.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself No funding was received for conducting this study.

Conflicts of Interest The authors have no conflicts of interest to declare.