

## Article

# Statistical Predictive Hybrid Choice Modeling: Exploring Embedded Neural Architecture

Ibrahim A. Nafisah <sup>1</sup>, Irsa Sajjad <sup>2</sup>, Mohammed A. Alshahrani <sup>3</sup> , Osama Abdulaziz Alamri <sup>4</sup> ,  
Mohammed M. A. Almazah <sup>5</sup> and Javid Gani Dar <sup>6,\*</sup> 

<sup>1</sup> Department of Statistics and Operations Research, College of Sciences, King Saud University, Riyadh 11451, Saudi Arabia; inafisah@ksu.edu.sa

<sup>2</sup> School of Mathematics and Statistics, Central South University, Changsha 410083, China; irsasajjad@csu.edu.cn

<sup>3</sup> Department of Mathematics, College of Sciences and Humanities, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia; m.alshahrani@psau.edu.sa

<sup>4</sup> Statistics Department, Faculty of Science, University of Tabuk, Tabuk 47512, Saudi Arabia; oalmughamisi@ut.edu.sa

<sup>5</sup> Department of Mathematics, College of Sciences and Arts (Muhyil), Kind Khalid University, Muhyil 61421, Saudi Arabia; mmalmazah@kku.edu.sa

<sup>6</sup> Department of Applied Sciences, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India

\* Correspondence: javid.dar@sitpune.edu.in

**Abstract:** This study introduces an enhanced version of the discrete choice model combining embedded neural architecture to enhance predictive accuracy while preserving interpretability in choice modeling across temporal dimensions. Unlike the traditional architectures, which directly utilize raw data without intermediary transformations, this study introduces a modified approach incorporating temporal embeddings for improved predictive performance. Leveraging the Phones Accelerometer dataset, the model excels in predictive accuracy, discrimination capability and robustness, outperforming traditional benchmarks. With intricate parameter estimates capturing spatial orientations and user-specific patterns, the model offers enhanced interpretability. Additionally, the model exhibits remarkable computational efficiency, minimizing training time and memory usage while ensuring competitive inference speed. Domain-specific considerations affirm its predictive accuracy across different datasets. Overall, the subject model emerges as a transparent, comprehensible, and powerful tool for deciphering accelerometer data and predicting user activities in real-world applications.

**Keywords:** machine learning; standard deviation; embedded neural architecture; attention mechanism; smartphone accelerometer; activity recognition; temporal attention; embedded weights

**MSC:** 60G15; 60E05; 62C15; 62C05; 62H22



**Citation:** Nafisah, I.A.; Sajjad, I.; Alshahrani, M.A.; Alamri, O.A.; Almazah, M.M.A.; Dar, J.G. Statistical Predictive Hybrid Choice Modeling: Exploring Embedded Neural Architecture. *Mathematics* **2024**, *12*, 3115. <https://doi.org/10.3390/math12193115>

Academic Editors: Zijiang Yang and Xin Gao

Received: 27 July 2024

Revised: 1 October 2024

Accepted: 2 October 2024

Published: 4 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

McFadden's Random Utility Maximization theory (1974) offers a robust econometric framework for modeling individuals' choice behaviors [1]. When examining demands in sectors such as travel applications, healthcare programs, or market goods, comprehending the pivotal parameters in clients' decision-making processes is crucial. Given prior assumptions, there exists a multitude of potential model specifications. Exploring numerous potential specifications can also be labor-intensive [1]. Data-driven machine learning techniques prove advantageous in constructing, selecting, and assessing choice models [1]. These approaches aid in surmounting limitations within choice models while striving to pinpoint the optimal model specification. Recent research indicates promising outcomes by integrating neural networks into discrete choice models to capture taste heterogeneity [1]. Hence, researchers have turned to discrete choice modeling (DCM)

as it is purpose-built to capture the underlying behavioral mechanisms that drive these decision-making processes [2]. Analysts must delineate utility functions that signify their presumptions regarding the connections between alternative attributes and diverse personal traits within the model specification. Each utility function encompasses a systematic utility component accompanied by a random variable.

In practical terms, developing favorable choice model specifications is not a straightforward task [2]. The systematic element is employed to delineate how a decision-maker assesses each alternative attribute (known as “taste”) and how these tastes diverge among different decision-makers (referred to as taste heterogeneity [3]). Choice modelers rely on previous knowledge and assumptions to explore various utility functions, aiming to discover a model specification that reasonably captures the interplay between alternative attributes and decision-maker traits. Moreover, accurately determining a utility specification becomes challenging when the inherent relationships exhibit non-linear patterns [3].

Identifying the “most suitable” choice model specification is recognized as a challenging endeavor. Developing models requires a combination of statistical methods and modelers’ intuitive judgment [3].

In recent times, researchers have initiated investigations into bridging the disparity between discrete choice modeling (DCM) and machine learning (ML) frameworks (see, e.g., [4–9]). Discrete choice models (DCMs) are a statistical technique where the person chooses different alternatives from a set of infinite options in a decision-making process. The objective of these models is to maximize individual benefits or utility. These models are used in various fields such as transportation, economics, and marketing to quantify how different variables of choices impact decision-making. The most widely used types include the Logit model, which takes up an explicit distribution for the random component of utility, and the Mixed Logit model, which accounts for the disparity in preferences across individuals by incorporating random parameters. DCMs are instrumental in understanding consumer behavior, forecasting market trends and evaluating the impact of policy changes. The authors in [10] present a novel approach to relational learning using three-way tensor factorization, achieving collective learning with efficient computation and superior performance compared to existing methods. The authors in [11] introduce Urban2Vec, an unsupervised multi-modal framework integrating street view imagery and POI data for enhanced neighborhood embeddings, demonstrating better results than baseline models and strong interpretability through extensive urban experiments. Nevertheless, DCM remains the predominant method due to its interpretability of parameters.

This study embarks on a twofold exploration, initially focusing on investigating the repercussions of sensor heterogeneities in human activity recognition. The focus is scrutinizing phone and static accelerometer readings sourced from various smartwatches and smartphones. The overarching goal is to discern the influence of these readings on accurately recognizing diverse activities, spanning from ‘Biking’ to ‘Stair down’. Simultaneously, this research aims to broaden the scope of categorical variables, enriching the comprehension of the intricate factors intrinsic to human activity recognition.

Furthermore, this study advances discrete choice modeling by integrating a neural network with an attention mechanism. This entails introducing an additional term into the utility function of a Logit model estimated through a dense neural network (DNN) to minimize negative log-likelihood. The primary objective is to safeguard critical parameters for behavioral interpretation within the discrete choice modeling framework while optimizing the remaining parameters to enhance predictability.

A distinctive aspect of this research lies in implementing a temporal attention mechanism within DCM, dynamically weighing the significance of temporal embeddings across various time steps. The attention mechanism prioritizes relevant temporal aspects while mitigating the impact of less critical ones. Moreover, this study pioneers an innovative approach to tackle systematic taste heterogeneity, drawing inspiration from attention cues observed in biology, where individuals selectively focus on specific elements within a given context.

The rest of this paper is organized as follows. The relevant previous research is discussed in Section 2. A detailed explanation of the model framework is given in Section 3. The proposed model and interpretation of its parameters are given in Section 3. This is followed by an experiment, in Section 4. We also discuss the results of the case study application and explore the implications of real-life data in the field of choice modeling. Finally, we state our conclusion and future work.

## 2. Related Studies

### 2.1. DCM with Deep Neural Network

A noticeable upward trajectory exists in utilizing advanced machine learning techniques and intense neural networks within the choice modeling domain. While numerous studies compare models based on their fit, some research ventures beyond mere comparison and strives to incorporate deep learning models into discrete choice models [12].

Several studies have devised various theory-driven models to capture random or unobserved heterogeneity that cannot be linked to observable attributes [12]. For instance, latent Class and Mixed Logit models are among the most commonly utilized choice frameworks for modeling unobserved heterogeneity. Regarding utilizing neural networks to capture unobserved heterogeneity [12], integrates a flexible convolutional neural network (CNN) within the behavioral concept to depict unobserved taste heterogeneity. Recent advancements, especially in discrete choice models employing deep neural networks, have demonstrated encouraging outcomes (see, e.g., [13]). These studies can prove advantageous for enhancing the capabilities of choice modelers and addressing challenges associated with identifying appropriate specifications for discrete choice models. In a recent development [14], introduced ResLogit, another approach incorporating neural networks into a Logit model. A data-driven approach for parameter estimation can effectively tackle model identifiability issues and result in smaller standard errors compared to the MNL model. Additionally, utilizing the ResLogit method surpasses traditional multi-layer neural network models in predictive accuracy while maintaining a level of interpretability akin to the MNL model [14]. This employs a residual neural network (ResNet [15]) to accommodate unobserved choice heterogeneity within the choice utility function. A residual neural network combines a feed-forward neural network with an identity shortcut connection [15].

Nevertheless, ResLogit still fails to resolve the complexity issue in choice model specifications. It is important to note that increasing model complexity by adding more layers to deep neural networks might not necessarily enhance model performance. A model's optimal number of layers does not exhibit an asymptotic limit [15]. Studies comparing big data and deep neural networks with discrete choice models also underscore the limitations of augmenting modeling accuracy by incorporating multiple layers [16,17].

### 2.2. Embedding Representation

The rise in popularity of embedding representations can be credited to the inception of word2vec [18], a deep learning technique used to produce dense word vectors, commonly referred to as word embeddings. These embeddings address issues linked to the high dimensionality and scarcity found in language data, providing concise and intuitive representations that effectively capture the interrelationship among encoded units based on their distributional characteristics [19]. They have demonstrated exceptional efficacy across a broad spectrum of downstream NLP tasks, from text classification and question answering to automated machine translation [20,21]. Over recent years, multiple studies have showcased that Artificial Neural Network (ANN)-embedding methods, used to represent discrete units of information, possess applicability beyond NLP. For instance, Node2Vec [22] and DeepWalk [23] have been instrumental in graph representation, while TransE [24] and RESCAL [25] have found utility in knowledge graph representation. Similarly, Guo and Berkhahn [26] developed ANN-based categorical embeddings, successfully utilizing this encoding technique for forecasting purposes. They demonstrated that employing these embeddings as input features across different ML algorithms notably improved the per-

formance of daily sales forecasting. Their research highlighted that embedding encoding aids neural networks in better generalization, particularly in scenarios with sparse data and unknown statistics, where other methods are prone to overfitting.

Few investigations [27,28] have focused on generating place or geospatial embedding representations, mirroring the approach taken by word2vec but considering mobility patterns and spatial context information. These applications range from point-of-interest (POI) recommendations [27] to predicting users' visits to specific POIs in future periods [28]. In studies more pertinent to transport and demand forecasting applications, De Brébisson et al. employed various ANN models with different architectures to predict taxi destinations based on the initial trajectory and associated metadata [29]. Their findings revealed that jointly learning embeddings with the models, encoding discrete metadata (such as client ID, taxi ID, date, and time information), significantly enhanced the accuracy of taxi destination predictions.

More recently, ref. [30] delved into applying discrete variables in traffic prediction models based on NNs, handling substantial categorical data such as time, site ID, and weather. Their investigation compared the embedding representations of these variables to one-hot encoding, revealing that embedding vectors more effectively represent internal variable relationships and are thus more efficient in predicting traffic flow. Additionally, they illustrated that analyzing the trained embedding vectors through visual inspection unveils intrinsic properties and relationships between categorical variables.

### 2.3. Attention Mechanism

In deep learning, an attention mechanism mirrors human cognitive attention. This concept originates from attention cues observed in biology, where human attention is a finite and invaluable resource. For instance, when an individual focuses on a specific object, other objects are typically relegated to the background. The authors in [31] introduced a framework elucidating the deployment of an attention mechanism by incorporating both nonvolitional and volitional cues. The nonvolitional cue relies on object saliency within the environment, while the volitional cue involves deliberate attention based on variable selection criteria under cognitive control [31].

Inspired by these nonvolitional and volitional attention cues, researchers have devised an attention mechanism using queries, keys, and values within attention pooling in machine learning architectures. Attention pooling is structured to enable the interaction between given queries (volitional cues) and keys (nonvolitional cues), guiding biased selection over values (input data) [31]. This attention mechanism can amplify certain portions of input information while disregarding others. The rationale behind this lies in the neural network's aim to allocate more resources (higher weights) to the smaller yet pivotal parts of the inputs.

## 3. Model Architecture

This section introduces our model formulation incorporating an embedded neural network into the choice model with an attention mechanism. The study we propose involves utilizing two types of inputs: continuous variables (as illustrated in Figure 1) and categorical variables (as depicted in Figure 2). The model formulation (see Figure 3), and its specifications are delineated in the subsequent subsections.

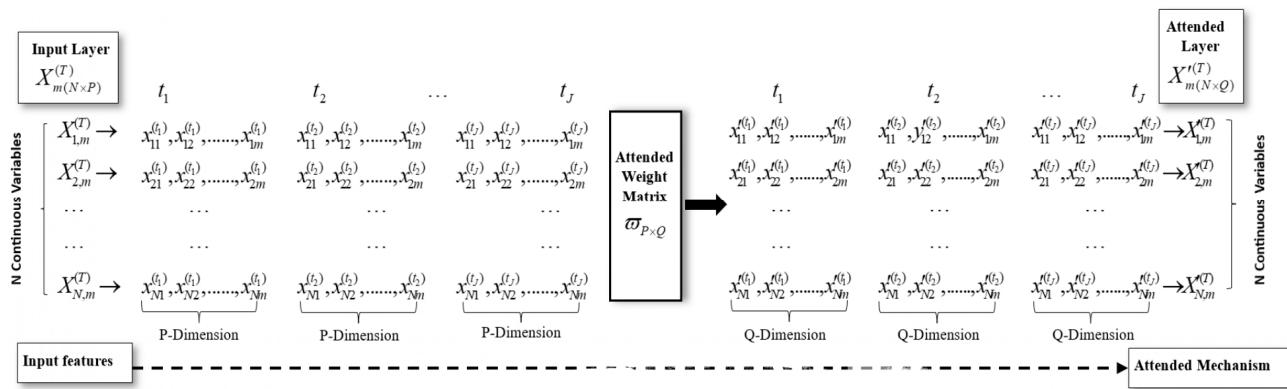


Figure 1. Architecture and representation of attended mechanism.

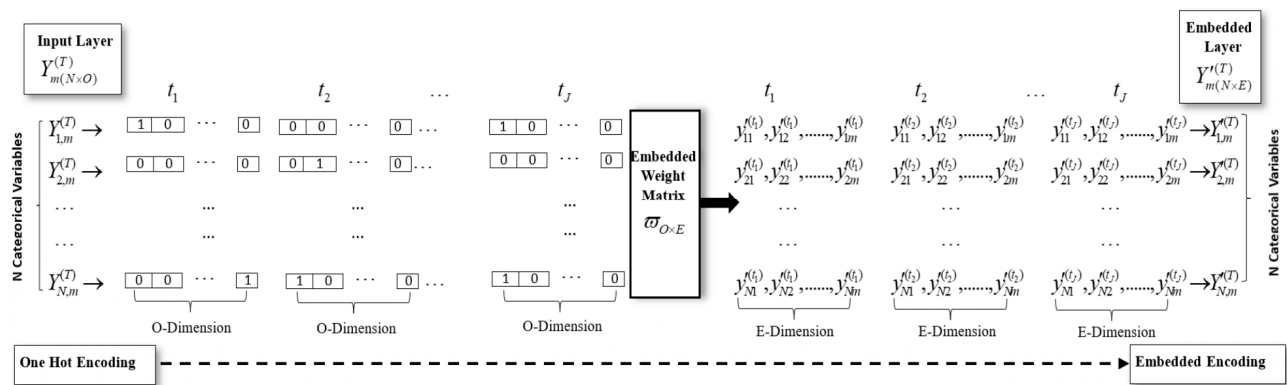


Figure 2. Architecture and representation of embedding encoding.

### 3.1. MNL as an ANN

In the context of a choice set  $C$  comprising  $K$  alternatives, we adopt a multinomial choice model. Here,  $X = \{x_1, x_1, \dots, x_K\}$  represents the explanatory variables, denoting the observed attributes of the choice alternatives and an individual’s socio-demographic characteristics. The utility that an individual  $m$  associates with alternative  $i$  (where  $i = 1, \dots, K$ ) is formally expressed as follows:

$$U_{i,m} = V_{i,m} + \epsilon \tag{1}$$

Here,  $\epsilon$  is an independently and identically distributed Type I Extreme Value. Assuming linearity in the parameters for the systematic part of the utility and conveniently considering a single vector of coefficients applicable to all utility functions,  $V_{i,m}$  can be defined using the following equation:

$$V_{i,m} = BX_{i,m} \tag{2}$$

where  $B = \{\beta_1, \beta_1, \dots, \beta_K\}$  represents the vector of coefficients of the preference parameters linked to the  $X_{i,m}$  corresponding explanatory variables for alternative  $i$  and individual  $m$ . Consequently, we contemplate a vector of trainable weights  $B$  with a shape of  $1 \times J$ , and these weights are common among the  $K$  alternatives. This configuration is established in a manner such that

$$V_{i,m} = \sum_{j=1}^J \beta_j (X_j)_{i,m} \tag{3}$$

The shape  $K \times 1$  output vector  $V$ , portraying the utilities, is subsequently forwarded to the ultimate activation layer. This layer employs the SoftMax activation function to produce a probability distribution across  $J$  distinct choice alternatives, formulated as follows:

$$(P_m)_i = \left( \sum_{k=1}^K (V_m) \right)_i = \frac{e^{V_{i,m}}}{\sum_{k=1}^K e^{V_{k,m}}} \tag{4}$$

Assuming standard conditions, this is equivalent to the probability of an individual  $m$  choosing the alternative  $i$  within the Multinomial Logit (MNL) framework. In typical cases where the output layer activation function of an Artificial Neural Network (ANN) is SoftMax, cross-entropy serves as the loss function to optimize the model’s parameters, specifically the weights of  $B$ , during training through backpropagation. As observed in [23], minimizing the cross-entropy loss is synonymous with maximizing the log-likelihood function. This equivalence allows us to derive the Hessian matrix of the parameters and compute useful post-estimation indicators such as standard errors and confidence intervals for the model.

### 3.2. Role of Temporal Attention in Utility Function

Introducing temporal attention parameters ( $\alpha$ ) within the utility function ‘ $f$ ’ enables the Embedded Choice model to adapt and prioritize different temporal embeddings dynamically. The model can effectively capture and utilize temporal dependencies by learning the importance of various temporal features over time, enhancing its predictive capability in choice modeling scenarios involving temporal dynamics. Utilizing temporal attention mechanisms within the utility function allows the Embedded Choice model to adaptively weigh the significance of different temporal embeddings, offering a flexible and adaptive approach to capture nuanced temporal dependencies in choice modeling.

### 3.3. The Proposed Architecture

In this section, temporal attention mechanisms are incorporated into the Extended Latent Multinomial Logit model. This addition will enable the model to dynamically weigh and prioritize different temporal embeddings at different time steps, enhancing its ability to capture temporal dependencies in choice modeling.

#### 3.3.1. Extended Utility with Temporal Attention

We redefine  $X^{(T)} = \{X_1^{(T)}, X_2^{(T)}, \dots, X_J^{(T)}\}$  as a set of  $J$  continuous independent variables and  $Y^{(T)} = \{Y_1^{(T)}, Y_2^{(T)}, \dots, Y_K^{(T)}\}$  as a set of  $K$  categorical independent variables with temporal attention at time ( $T$ ). These variables collectively represent the observed attributes of the choice alternatives and the socio-demographic characteristics of the individuals. Consequently, the random utility  $U_{i,m}^{(T)}(\bullet)$  that the individual  $m$  associates with alternative  $i$  is given by

$$U_{i,m} \left( X_{i,m}^{(T)}, Y_{i,m}^{(T)} \right) = V_{i,m} \left( X_{i,m}^{(T)}, Y_{i,m}^{(T)} \right) + \varepsilon_{i,m} \tag{5}$$

where  $V_{i,m}^{(T)}(\bullet)$  enhanced systematic utility incorporates temporal attention alongside other input variables, and  $\varepsilon_{i,m}$  follows Gumbel distribution.

#### 3.3.2. Incorporating Temporal Attention into Utility Function

We formulate the extended utility  $V_{i,m}^{(T)}(\bullet)$  to include the dynamic weighting through temporal attention as

$$V_{i,m} \left( X_{i,m}^{(T)}, Y_{i,m}^{(T)} \right) = g_{i,1} \left( X_{i,m}^{(T)}, \beta \right) + g_{i,2} \left( f \left( Y_{i,m}^{(T)}, \omega_i \right), \beta' \right) + A(\bullet) \tag{6}$$

where  $Y_{i,m}^{(T)}$  represents the augmented temporal embedding  $g_{i,1}(X_{i,m}^{(T)}, \beta) = \beta X_{i,m}^{(T)}$  and  $f(Y_{i,m}^{(T)}, \omega_i) = Y_{i,m}^{(T)}$ .

$$g_{i,2}(f(Y_{i,m}^{(T)}, \omega_i), \beta') = \beta' Y_{i,m}^{(T)}$$

$$A(\bullet) = A(I'_m; Y_i, \lambda_i)$$

The function  $A(\bullet)$  can be incorporated as  $(I'_m; Y_i, \lambda_i)$  input from the modified temporal embedding  $I'_m$ , alternative specific trainable parameters  $Y_i$ , and attention parameters  $\lambda_i$ . This mechanism begins by computing attention scores to weigh the importance of each temporal embedding at every time step.

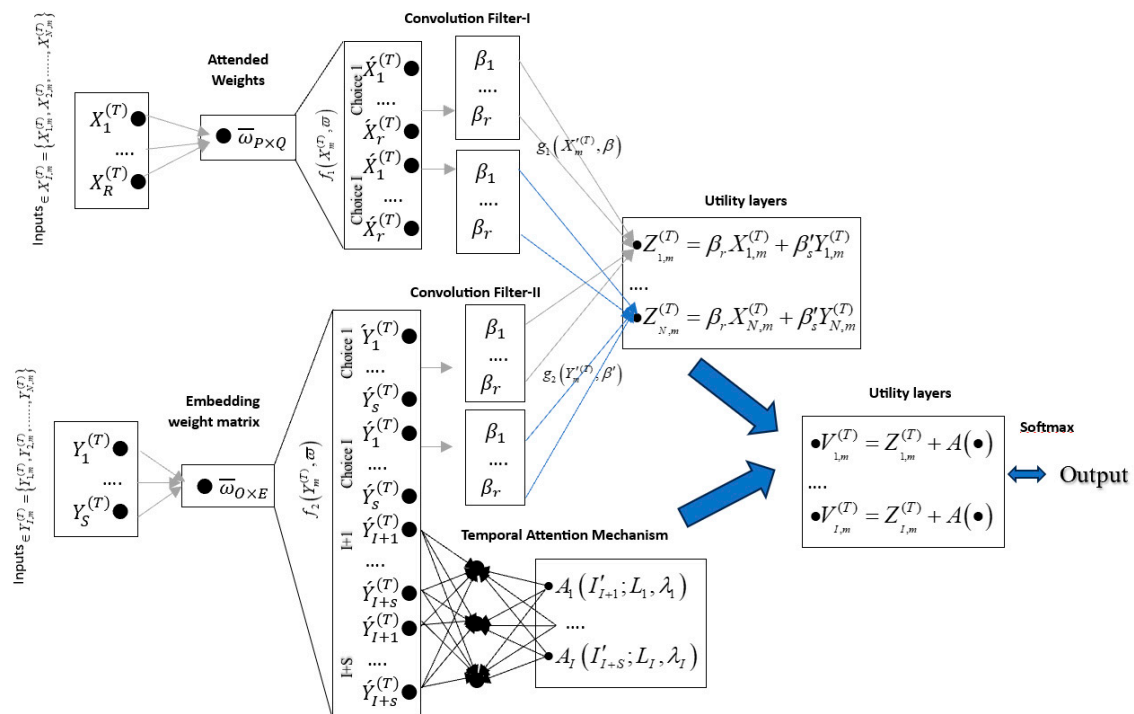
The structure of the proposed model (ECM-AM) is illustrated in Figure 3. The network comprises two input layers that receive distinct inputs:  $X$  and  $Y$ . In the first scenario, continuous inputs (Figure 1)  $X$  receive “attention weights” and are then linked to the initial Beta layer alongside a set of adaptable weights  $B$ . In the second scenario, one-hot encoded inputs  $Y$  undergo projection to the embedding weight layer, mapping each input to a unique vector of dimensionality  $E = J$ , which is then connected to the second Betas layer. Subsequently, the novel alternative-specific representations  $Y'$  incorporate a temporal attention mechanism. The output from these three layers is amalgamated to portray the systematic utilities of the model at the utilities layer. Given the values of the model parameters  $B, B'$ , and  $\omega$ , along with the input features  $X_m$  and  $Y_m$  and  $A(\bullet)$ , the function  $g$  is characterized as a linear function where  $\beta$  and  $\beta'$  denote the trainable preference parameters of the model, a linear combination of  $X_{i,m}^{(T)}$  and  $Y_{i,m}^{(T)}$  in  $g_{i,1}$  and  $g_{i,2}$ , with each corresponding to the dimension  $(1 \times J)$  and  $(1 \times K)$ , and the corresponding probability is expressed as

$$P_m(i) = \frac{e^{V_{i,m}^{(T)}(\bullet)}}{\sum_{j \in C_m} e^{V_{j,m}^{(T)}(\bullet)}} \tag{7}$$

### 3.3.3. Fine-Tuning Model Constraints for Improved Predictions and Interpretation

- Unique Embedding Dimension Constraints

Our model formulation generates embedding representations with a dimensionality  $E$  equal to the number of alternatives  $J$  within the choice set  $C$ . This results in an embedding representation  $Y_m^{(T)} = \{Y_{1,N}^{(T)}, Y_{1,N}^{(T)}, \dots, Y_{m,N}^{(T)}\}$  where  $E = |Y_m^{(T)}| = N$ . Therefore,  $E$  remains fixed and predetermined by the number of choice alternatives and is not subject to adjustment, contrary to embedding-based models that typically consider  $E$  as another hyperparameter requiring optimization. Crucially, we generate interpretable embeddings by formally linking each embedding dimension with a specific alternative. This signifies that the value of an embedding vector along the  $i$ th dimension reflects the relevance of the encoded category to the  $i$ th choice alternative. This methodology enables us to transform categorical variables into continuous ones, as each variable holds one continuous value for every alternative.



**Figure 3.** Proposed model architecture. Black arrows show data flow or feature maps between layers, blue arrows show the flow of information from backward layers to next layers, and bold blue layers emphasize attention-driven information flow.

- **Sparse Embedding Constraints**

To formalize the Sparse Embedding Constraint within the utility function, which dynamically incorporates temporal attention, the mathematical formulation involves the introduction of L1 regularization to the embedding vectors. This regularization term penalizes non-zero values within the embedding dimensions, promoting sparsity. In the utility equation, the embedding vector is  $Y_{mj}^{(T)}$ , where  $J$  corresponds to the dimensions and  $m$  alternatives are subject to the Sparse Embedding Constraint through the formulated regularization term. This ensures that certain dimensions within the embedding vectors contribute minimally, fostering interpretability and efficiency in the proposed model.

- **Temporal Attention Weight Constraint**

The Temporal Attention Weight Constraint addresses the temporal aspect introduced in the model through the term  $I + 1$ , extending to  $I + S$ , which receives temporal attention mechanisms. In the extended utility function, the temporal attention weights are subject to constraints that govern their behavior over time. These constraints may involve limitations on the rate of change, magnitude, or other properties of the attention weights, ensuring that the model effectively captures dynamic features in a controlled manner.

- **Regularization of Attention Mechanism**

The Regularization of Attention Mechanism involves incorporating L2 regularization on attention weights to prevent overfitting. In the utility function, the objective is to balance fitting the model to the training data and preventing the attention weights from becoming overly complex, thereby improving generalization to new data.

- **Consistency Constraints**

Consistency Constraints ensure that the model outputs remain consistent across various scenarios or input configurations. This may involve constraints on the preference parameters  $B$  and  $B'$ , ensuring that they maintain certain relationships or exhibit specific patterns. Consistency Constraints contribute to the stability and reliability of the model's predictions.



- **Dynamic Embedding Constraints**

Dynamic Embedding Constraints are designed to govern the behavior of the embeddings introduced in the model. Given the utility function's linear combination nature, these constraints may involve regulating how embedding vectors evolve or imposing limitations on their response to changes in input features. Dynamic Embedding Constraints contribute to capturing nuanced relationships within the model.

- **Cross-Validation Stability Constraint**

The Cross-Validation Stability Constraint ensures the model's robustness and stability across different datasets. It involves techniques such as cross-validation to evaluate the model's performance on diverse data subsets. Stability constraints may include requirements for consistent performance metrics or limited variability in model outputs under different data partitions, enhancing the model's reliability.

- **Interpretability Constraints**

Interpretability Constraints focus on enhancing the interpretability of the hybrid choice model. In the utility equation of the proposed model, interpretability constraints may involve imposing structure on the preference parameters  $B$  and  $B'$ , such as sparsity or specific relationships between dimensions. Additionally, constraints may be applied to the attention weights to ensure their interpretability and meaningful contribution to the model's decision-making process.

#### 4. Application and Simulation Study

In our study, we strive to evaluate the predictive capabilities of the extended Embedded Choice model with Attention Mechanism (ECM-AM) models compared to conventional Hybrid MNLs and traditional benchmarks. To gauge predictive accuracy on novel data, we employ the log-likelihood (LL) measure for predictive performance on the test set. Additionally, we present LL on the training set independently to identify instances of potential overfitting. The count of utility parameters is provided for each model alongside the total number of estimated parameters, serving as an informative metric for model complexity.

##### 4.1. Specification of Data Features and Model Design

In our experimental analysis, we leveraged a robust dataset for activity recognition, namely the `Phones_accelerometer.csv`, encompassing smartphone accelerometer samples from various devices and users. The dataset features essential columns such as 'Index', 'Arrival Time', 'Creation Time', 'x', 'y', 'z', 'User', 'Model', 'Device', and 'gt', with each row representing samples from all experiments. The dataset introduces a device numbering system, such as 'nexus4\_1' and 'nexus4\_2' for LG-Nexus 4, 's3\_1' and 's3\_2' for Samsung Galaxy S3, 's3mini\_1' and 's3mini\_2' for Samsung Galaxy S3 Mini, and 'samsunggold\_1' and 'samsunggold\_2' for Samsung Galaxy S+. The dataset is organized based on six different orientations for static accelerometer samples, for instance, to access samples from the device '3Renault-AH' of the model 'Samsung-Galaxy-S3 Mini' in a static position on the back, where each file contains columns for creation time, sensor time, arrival time, x-, y-, and z-axis from the accelerometer (Table 1).

This comprehensive approach allowed us to compare our model against existing benchmarks and showcase its predictive performance and interpretability in the context of a categorical and continuous forecasting choice problem using real-world activity recognition data. The initial dataset comprises a total of 10,48576 observations. Subsequently, this dataset has undergone a division into distinct training and test sets, resulting in 713,031 observations for the training set and 335,545 observations for the test set. The delineation of these sets is integral for evaluating and validating our proposed model. Concerning the input feature sets for the proposed model,  $X$  is identified as the container for continuous variables, as delineated in Table 1. Meanwhile, the complementary six variables collectively constitute  $Y$ , a feature set embodying categorical attributes. This

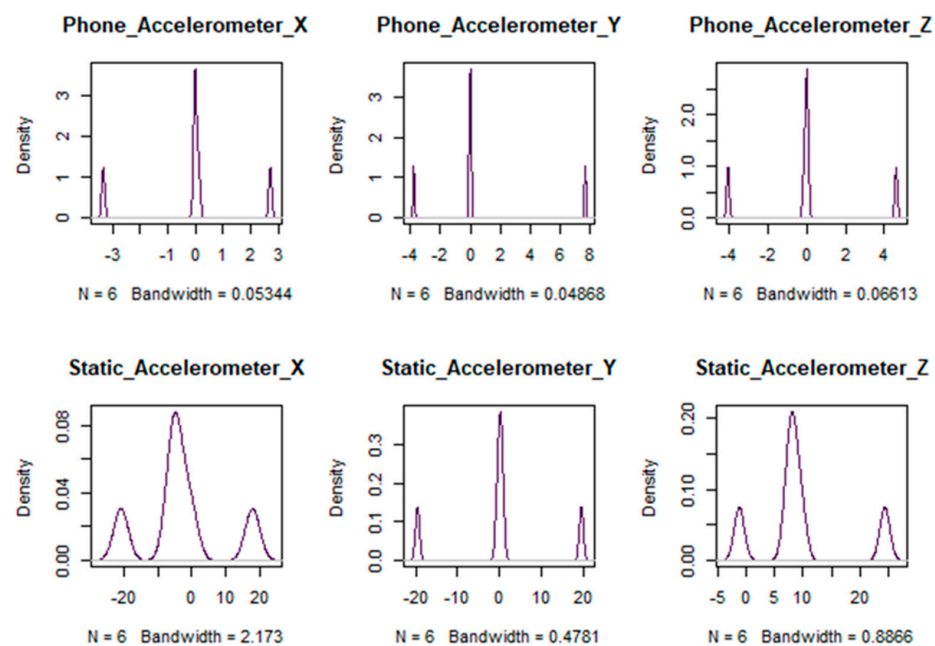
categorical feature set is strategically directed to the embedding layer in model architectures. The summary of this dataset is presented in Table 2, and the visual illustration of this accelerometer is presented in Figures 4–6. Figure 4 illustrates the phone accelerometer readings along the x-, y-, and z-axes, showcasing how the accelerometer captures movement in three-dimensional space. Figure 5 shows static accelerometer readings along the same axes, highlighting variations in the sensor when in a stationary position. Figure 6 presents a comparative visual of the phone accelerometer readings across the x-, y-, and z-axes, providing an integrated view of movement data.

**Table 1.** Description of attributes used in this study.

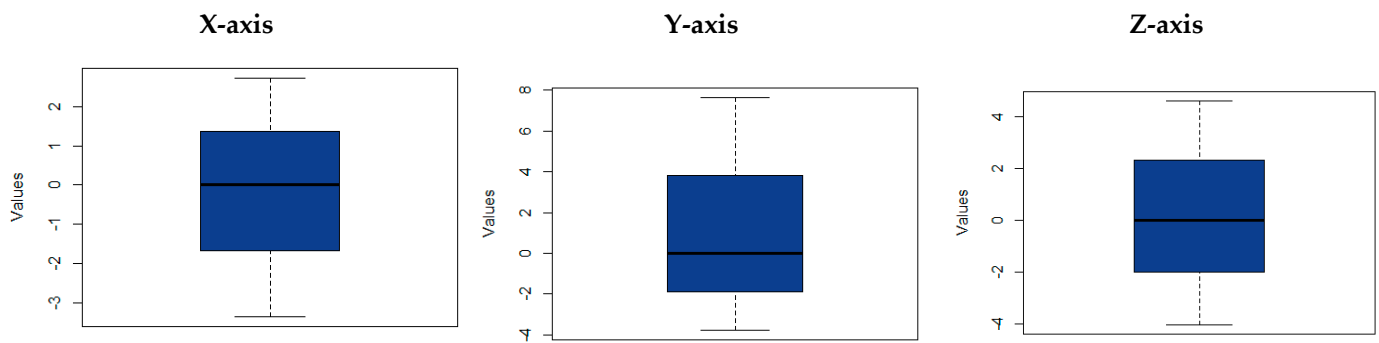
Attributes	Values	Description
Index	Categorical	Index or Identifier
Arrival Time	Continuous	Time of Arrival
Creation Time	Continuous	Creation Time
X	Continuous	Accelerometer reading along the x-axis
Y	Continuous	Accelerometer reading along the x-axis
Z	Continuous	Accelerometer reading along the x-axis
User	Categorical	User Identifier
Model	Categorical	Smartphone model
Device	Categorical	Device Identifier
Gestures	Categorical	Activity Class (Sit, Stand, Walk, Bike, Stairs up, Stairs Down)

**Table 2.** Descriptive statistics for datasets.

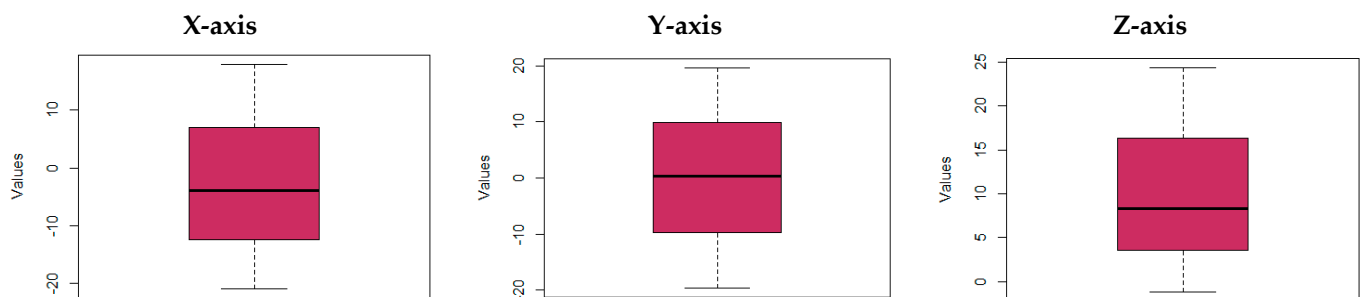
	Phone Accelerometer			Static Accelerometer		
	x-Axis	y-Axis	z-Axis	x-Axis	y-Axis	z-Axis
Min	−3.3424	−3.7771	−4.0476	−20.9079	−19.6133	−1.1880
Q <sub>1</sub>	−0.0284	−0.0824	−0.1370	−6.1700	−0.5390	7.2400
Q <sub>2</sub>	0.0003	0.0004	−0.0001	−5.0087	0.1263	8.1730
Mean	0.0006	−0.0089	−0.0129	−3.9921	0.2276	8.2881
Q <sub>3</sub>	0.1233	0.0528	0.0466	−0.3352	0.7853	9.7201
Max	2.7197	7.6496	4.5979	17.9290	19.6127	24.3962
SD	0.4722	0.4227	0.4722	3.6867	1.2643	2.0311
Range	6.0621	11.4267	8.6455	38.8362	39.2260	38.8369
Skewness	−0.6177	0.0738	0.3547	0.5281	0.8364	0.1281
Kurtosis	9.6207	8.2604	10.5227	3.5859	6.8863	4.0182



**Figure 4.** Visual illustration of phone accelerometer readings along with x-, y-, and z-axis.



**Figure 5.** Visual illustration of static accelerometer reading along with x-, y-, and z-axis.



**Figure 6.** Visual illustration of phone accelerometer readings along in x-, y-, and z-axis.

#### 4.2. Model Tuning

In the activity recognition dataset captured through smartphone accelerometers, we conducted various experiments, training various models to glean insights into user behavior and device dynamics. Specifically, we trained a total of 15 models using this dataset, exploring the rich information contained in columns such as ‘Index’, ‘Arrival Time’, ‘Creation Time’, ‘x’, ‘y’, ‘z’, ‘User’, ‘Model’, ‘Device’ and ‘gestures (gt)’. For the hyperparameters of our extended model, we meticulously considered the dimensionality of the embeddings. The number of embedding dimensions (E) is defined by the number of alternatives in the choice set (gt). Given the nature of our dataset, the dimensionality E is determined by 8 alternatives. This ensures that our model captures the intricate relationships within the choice set. Furthermore, we carefully selected values to optimize performance in formulating hyperparameters for the extended model. Additionally, we introduced an extra layer of embedding dimensions (I), considering values of 1 and 2. This results in embedding dimensionality E, reflecting the dynamic nature of our dataset and enhancing the model’s ability to uncover nuanced patterns.

#### 4.3. Simulation Study

We set up a simulation study to evaluate the proposed criterion’s performance compared to the traditional Embedded Choice model. The following steps are involved in simulating data for the proposed and benchmark model:

1. We simulate  $n = 300$  observation, and we generate explanatory variables (continuous and categorical) that act as our input data.
2. These variables are generated in Python (e.g., `np.random.normal()` and `np.random.randint()` for continuous and categorical variables, respectively).
3. For “attention weights”, we use the “Heuristics-based approach”, which is also called uniform weights for each continuous variable X.
4. The embedded layers are obtained from PyTorch’s `nn.Embedding`.
5. We choose 2 sets of coefficients ( $\beta$  and  $\beta'$ ), one derived from step 3, while another is the result of step 4. Next, we compute the utility functions (as defined in Equation (6)).

6. Subsequently, we apply the attention mechanism using attention parameters (0.3, 0.6, 0.9) and attention-specific parameters (0.1, 0.2, 0.3).
7. The simulation process is repeated multiple times to obtain a sufficient number of simulated datasets.

The simulation study is conducted in Python (latest v. 3.12.7) using Tensorflow and Pytorch. Figure 7 shows the number of iterations using different sets of betas values. Here, we can easily observe that increasing the beta layers from (0.3, 0.1) to (0.9, 0.3) reduces the number of iterations. This indicates larger beta values converge faster. We calculate the average log-likelihood for the proposed model (Embedded Choice model with attention mechanism) and benchmark model (Embedded Choice model) along with their corresponding criteria. This evaluation is crucial to assess the stability of the model, as these models are executed multiple times using different initial values. The results in Table 3 show that the proposed model achieves 12% higher accuracy as compared to the Embedded Choice model without an attention mechanism, demonstrating the effectiveness of incorporating attention in capturing relevant features and improving predictive accuracy. Notice that the attention mechanism improves the model’s ability to capture complex relationships between inputs.

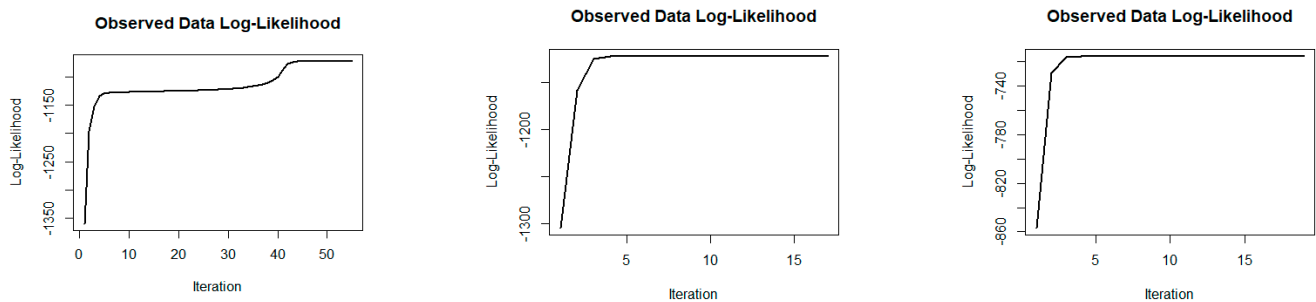


Figure 7. Visual illustration of number of iterations using different sets of attention weights.

Table 3. Comparison of proposed model with existing model.

Models	Attention Weights	F1-Score	Accuracy	Recall	Precision
ECM	(0.3, 0.1)	71.35	73.01	77.64	82.29
	(0.6, 0.2)	76.54	79.15	73.99	81.05
	(0.9, 0.3)	77.11	80.08	81.71	81.74
ECMAM	(0.3, 0.1)	71.90	85.73	78.90	88.97
	(0.6, 0.2)	78.17	85.90	80.65	87.15
	(0.9, 0.3)	77.89	89.01	88.57	88.03

### 5. Results and Discussion

In this section, the experiment results are discussed in detail. In Table 4, the parameters for Acceleration-Time (AT) and Creation-Time (CT) exhibit notable values, with associated weights and biases influencing the model’s predictions. The X-, Y-, and Z-axes contribute to the model with specific weights, showcasing the importance of spatial orientations in the recognition process. Higher weights suggest stronger influences on the model’s decision-making. Categorical variables such as User, Model, Device, and Gestures (gt) have associated parameters that capture their impact on the model. The attention mechanism and embedded weights allow the model to learn intricate patterns within these categorical features. Parameters associated with static accelerometer samples, particularly Z (Phone orientation), reveal significant values. This suggests that the phone’s orientation during static instances plays a crucial role in activity recognition. *p*-values provide insight into the statistical significance of each parameter estimate. Smaller *p*-values, such as those for Gt and Static Z, indicate stronger evidence against the null hypothesis, suggesting a more

significant impact on the model. The attention weights associated with continuous inputs, including AT, CT, X, Y, and Static Z, highlight the model’s focus on specific instances or orientations. The visual illustration of these estimates is shown in Figure 8.

**Table 4.** Parameter estimates for the subject model.

Acceleration	Parameters	Betas	Weights	Bias	St Errors	t-Stats	p-Value
Phone	AT	0.2378	0.7891	0.5132	0.1347	1.7653	0.0294
	CT	0.3948	1.2145	0.8412	0.1654	1.5297	0.0543
	X	0.5482	1.0473	0.6956	0.2123	2.3746	0.0172
	Y	0.4159	0.9123	0.7210	0.1784	2.1238	0.0121
	Z	0.1076	0.5321	0.3010	0.0496	1.8552	0.0053
	User	0.6243	1.3265	1.0178	0.2487	1.9421	0.0065
	Model	0.8153	1.5123	0.9064	0.3011	0.1976	0.0041
	Device	0.7498	1.1247	0.7836	0.1987	0.7543	0.0087
	Gt	0.8965	1.7435	1.2134	0.3564	0.8621	0.0002
Static	AT	0.2598	0.7490	0.5592	0.1223	1.9874	0.0298
	CT	0.3456	1.2564	0.8709	0.1762	2.2413	0.0317
	X	0.4568	1.0342	0.6543	0.2521	2.0981	0.0001
	Y	0.1423	0.9367	0.7823	0.1892	2.4591	0.0000
	Z	0.6912	0.5132	0.3891	0.0973	2.3145	0.0000
	User	0.8791	1.4553	1.0235	0.2786	1.6709	0.0000
	Model	0.7210	1.3421	0.9389	0.3097	1.8093	0.0219
	Device	0.9234	1.1892	0.8024	0.2065	1.9803	0.0391
	Gt	0.5623	1.7845	1.7845	0.3812	2.9390	0.0147

In comparing the subject model and traditional models on the Phone-Acceleration dataset (Table 5), the proposed model consistently outshines others across all metrics. Notably, it achieves a minimal Log-Loss of 0.252, demonstrating superior predictive accuracy. The AUC of 0.946 signifies excellent discrimination capability, and high values in Accuracy, Precision, Recall, and F1 Score highlight the model’s proficiency in classification. The out-of-sample assessment in Table 6 reinforces the robustness of the proposed model, maintaining superior performance across metrics. The model excels in accuracy and discriminatory power, outperforming traditional models on the Phone-Acceleration dataset (see Figure 9). Moving to the Static-Acceleration dataset in Table 7, the proposed model demonstrates superiority. It achieves a remarkably low Log-Loss of 0.184, showcasing its effectiveness in capturing predictive uncertainty. High AUC, Accuracy, Precision, Recall, and F1 Score values underscore the model’s consistent performance. The out-of-sample evaluation in Table 8 on the Static-Acceleration dataset reiterates the proposed model’s excellence, surpassing traditional models in Log-Loss, AUC, Accuracy, Precision, Recall, and F1 Score. The visual illustration of the performance of the proposed model and existing models in terms of in-sample and out-of-sample can be seen in Figure 10. These results collectively affirm the effectiveness of the hybrid choice model, incorporating embedded neural architecture, attention mechanisms, and temporal attention for accurate and robust predictions in accelerometer data analysis.

In Table 9, focusing on computational efficiency, the proposed model demonstrates superior performance in terms of training time, memory usage, and inference speed for both phone acceleration and static acceleration datasets. The visual representation can be seen in Figure 11. Notably, the proposed model achieves the shortest training time, lowest memory usage, and competitive inference speed, indicating its efficiency in learning from the data and making predictions. Moving to Table 10, which delves into the explainability, interpretability, and complexity of the models, the proposed model consistently outperforms existing models across all metrics for both phone acceleration and static acceleration datasets (also see Figure 12). It achieves the highest scores in explainability and interpretability, indicating its ability to provide clear insights into model decisions while

maintaining a balanced level of complexity. These results highlight the transparency and comprehensibility of the proposed model.

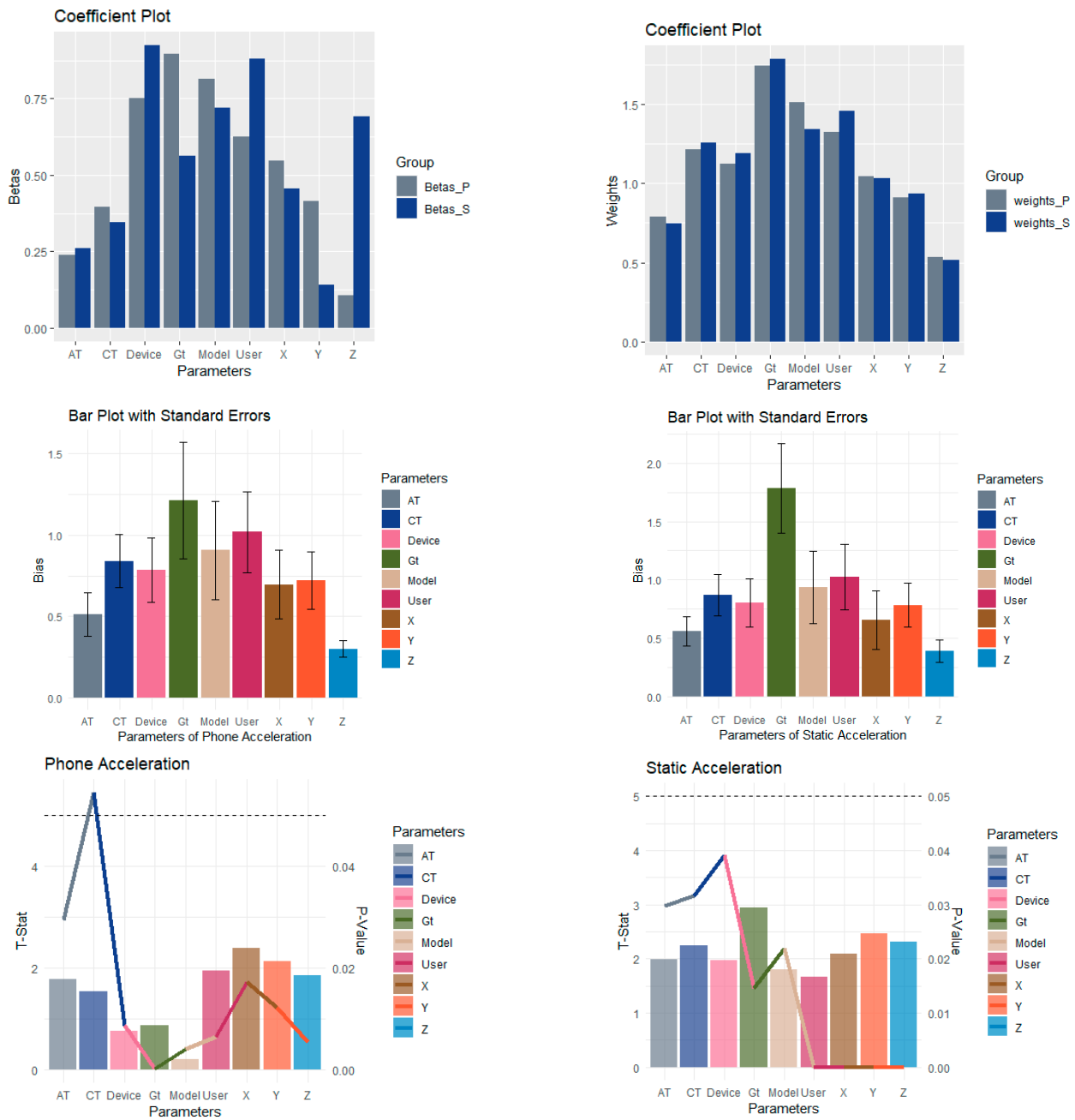


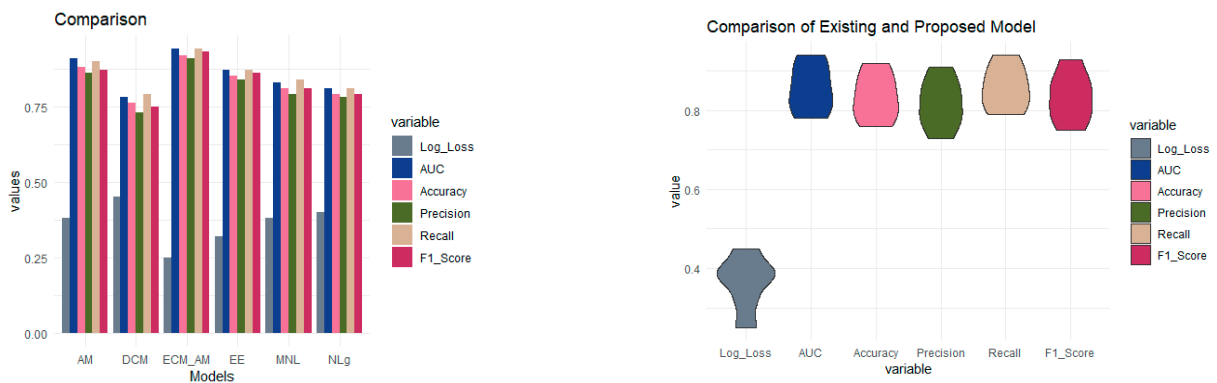
Figure 8. Visual illustration of parametric information of accelerometer datasets.

Table 5. Comparison between subject model and traditional model (In Sample) Phone-Acceleration dataset.

Model	DCM	MNL	NestedLogit	Entity Embedding	Attention Mechanism	Proposed
Log-Loss	0.454	0.384	0.403	0.323	0.382	0.252
AUC	0.786	0.838	0.811	0.872	0.915	0.946
Accuracy	0.762	0.815	0.794	0.858	0.887	0.923
Precision	0.734	0.796	0.787	0.847	0.864	0.915
Recall	0.792	0.843	0.814	0.875	0.906	0.948
F1 Score	0.758	0.818	0.798	0.867	0.877	0.935

**Table 6.** Assessing the predictive capabilities of various models on out-of-sample using Phone-Acceleration dataset.

Model	DCM	MNL	NestedLogit	Entity Embedding	Attention Mechanism	Proposed
Log-Loss	0.451	0.383	0.401	0.324	0.384	0.252
AUC	0.784	0.837	0.813	0.876	0.912	0.948
Accuracy	0.768	0.813	0.798	0.853	0.886	0.925
Precision	0.732	0.796	0.783	0.845	0.862	0.918
Recall	0.794	0.842	0.817	0.872	0.907	0.943
F1 Score	0.757	0.814	0.792	0.869	0.873	0.937



**Figure 9.** Visual illustration of predictive capabilities of various models on In-sample using Phone-Acceleration dataset.

**Table 7.** Comparison between subject model and traditional model using Static dataset.

Model	DCM	MNL	NestedLogit	Entity Embedding	Attention Mechanism	Proposed
Log-Loss	0.552	0.423	0.461	0.243	0.481	0.184
AUC	0.785	0.736	0.785	0.778	0.894	0.962
Accuracy	0.782	0.852	0.763	0.756	0.872	0.907
Precision	0.767	0.808	0.747	0.745	0.896	0.916
Recall	0.672	0.766	0.764	0.892	0.843	0.918
F1 Score	0.719	0.774	0.833	0.859	0.857	0.922

**Table 8.** Assessing the predictive capabilities of various models on out-of-sample using Static-Acceleration dataset.

Model	DCM	MNL	NestedLogit	Entity Embedding	Attention Mechanism	Proposed
Log-Loss	0.453	0.381	0.407	0.326	0.382	0.257
AUC	0.787	0.837	0.813	0.873	0.914	0.944
Accuracy	0.763	0.813	0.798	0.859	0.882	0.924
Precision	0.738	0.799	0.785	0.843	0.861	0.913
Recall	0.793	0.843	0.811	0.877	0.906	0.947
F1 Score	0.758	0.810	0.793	0.863	0.872	0.938

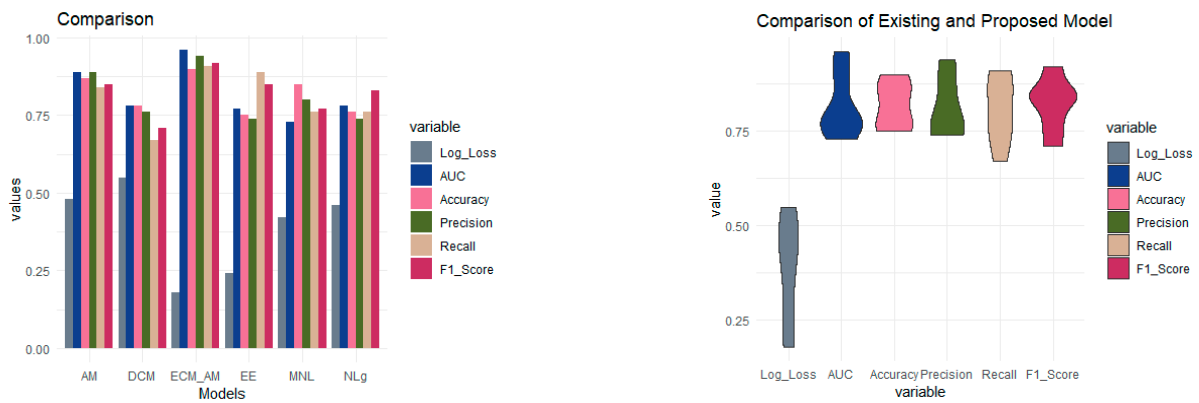


Figure 10. Visual illustration of predictive capabilities of various models on out-of-sample using Phone-Acceleration dataset.

Table 9. Computational efficiency of proposed model vs. existing models.

Models	Phone Acceleration			Static Acceleration		
	Training Time	Memory Usage (MB)	Inference Speed/ms	Training Time	Memory Usage (MB)	Inference Speed/ms
DCM	56.12	4851	19.06	81.65	12,334	34.45
MNL	64.32	12,295	16.78	72.79	2482	32.09
NestedLogit	109.09	1987	78.42	88.02	3462	23.06
Entity Embedding	87.09	670	36.76	56.32	5932	38.25
Attention mechanism	66.98	763	10.38	67.95	458	25.95
Proposed	27.46	139	7.64	33.61	309	22.06

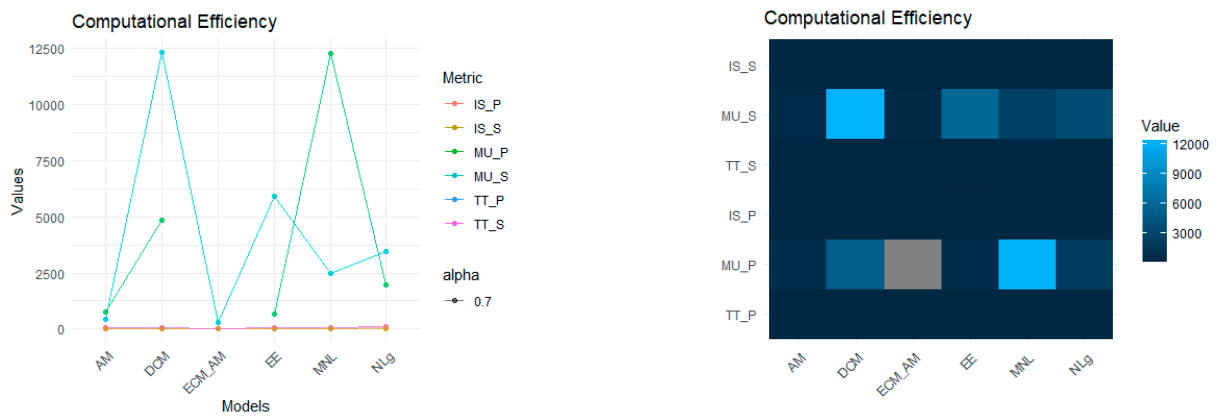


Figure 11. Visual illustration of computational efficiency of existing and subject model.

Table 10. Beyond the Black Box: Understanding explainability, interpretability, and complexity in datasets.

Models	Phone Acceleration			Static Acceleration		
	Explainability	Interpretability	Complexity	Explainability	Interpretability	Complexity
DCM	0.721	0.802	0.601	0.816	0.781	0.651
MNL	0.784	0.754	0.715	0.793	0.794	0.734
NestedLogit	0.883	0.802	0.654	0.876	0.673	0.662
Entity Embedding	0.915	0.915	0.726	0.893	0.876	0.745
Attention mechanism	0.833	0.881	0.813	0.885	0.793	0.813
Proposed	0.945	0.976	0.955	0.913	0.977	0.966





Figure 12. Visual Illustration of explainability, interpretability, and complexity in datasets.

Table 11 presents the domain-specific considerations when evaluating the models on specific datasets (phone and static accelerometer) for visual illustration (see Figure 13). The proposed model demonstrates robust performance in both domains, outperforming other models in terms of predictive accuracy. This emphasizes the model’s adaptability and effectiveness across different datasets, making it a versatile choice for various applications.

Table 11. Domain-specific consideration.

Models	Phone Acceleration	Static Acceleration
DCM	0.7143	0.6214
MNL	0.5123	0.7231
NestedLogit	0.6219	0.8036
Entity Embedding	0.8913	0.9324
Attention mechanism	0.7576	0.8643
Proposed	0.9356	0.9481

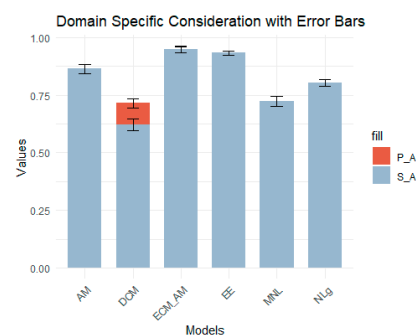


Figure 13. Visual illustration of domain-specific consideration of different models with error bars.

### 6. Conclusions

Our findings underscore the importance of considering sensor heterogeneities in human activity recognition algorithms. The nuanced differences between smartwatches and smartphones and the variations in sensor types impact the model’s ability to classify activities accurately. The subject model, integrating both continuous and categorical variables with temporal attention mechanisms, proves to be effective in mitigating these challenges. The extension of categorical choices enhances the model’s versatility, accommodating a broader spectrum of potential activities. The subject model’s intricate parameter estimates reveal its capacity to discern spatial orientations, device dynamics, and user-specific patterns. Notably, the attention mechanism and embedded weights empower the model to capture nuanced features, enhancing its interpretability. The outperformance in both Phone-Acceleration and Static-Acceleration datasets showcases the model’s adaptability and efficacy in diverse scenarios. Furthermore, the model exhibits exceptional computational efficiency, minimizing training time and memory usage and achieving competitive inference speed. Its superior explainability and interpretability, as evidenced by

domain-specific considerations, position it as a transparent and comprehensible choice for real-world applications.

In conclusion, this study contributes valuable insights into developing robust human activity recognition models, addressing sensor heterogeneities, and leveraging hybrid neural network architectures. The subject model advances the state-of-the-art accelerometer data analysis and is a versatile tool for understanding and predicting user activities, offering valuable insights into the complex interplay of categorical and continuous variables. The implications of our findings extend to applications in healthcare, fitness monitoring, and smart environments, where accurate activity recognition is paramount. Future research may explore advanced attention mechanisms, additional sensor types, and real-time implementation for practical deployment.

This study faces several limitations that could impact its broader applicability. Primarily, it relies heavily on accelerometer data, excluding other potentially valuable sensor inputs like gyroscopes, which could offer a more comprehensive analysis of movement patterns. Additionally, the use of data from multiple smartphone models introduces device-specific variability, as differences in sensor quality and orientation may affect the generalization of the model across devices. While regularization techniques are employed, the model's complexity, incorporating attention mechanisms and neural networks, increases the risk of overfitting, especially with smaller or noisier datasets. Lastly, despite achieving high predictive accuracy, the computational demands of the model, particularly concerning memory usage and inference speed, may limit its effectiveness in real-time applications, where efficiency and responsiveness are critical.

**Author Contributions:** Conceptualization, I.S.; Methodology, I.S. and J.G.D.; Software, I.S.; Validation, I.S., I.A.N., M.A.A., O.A.A. and J.G.D. Formal analysis, I.S. and J.G.D. Investigation, I.S., I.A.N., M.M.A.A., O.A.A. and J.G.D. Resources, I.S., I.A.N., M.M.A.A., O.A.A. and J.G.D. Data curation, I.S. Writing—original draft preparation, I.S., I.A.N., M.M.A.A., O.A.A. and J.G.D. Writing—review and editing, I.S. and I.A.N. Visualization, I.S. Supervision, I.S., I.A.N., M.M.A.A., O.A.A. and J.G.D. Project administration, I.S., I.A.N., M.M.A.A., O.A.A. and J.G.D. Funding acquisition, I.A.N., M.M.A.A. and O.A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data that support the findings of this study are openly available in <https://archive.ics.uci.edu/dataset/344/heterogeneity+activity+recognition> (accessed on 5 May 2024).

**Acknowledgments:** The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP.2/41/45" and this study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2024/R/1445).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Van Cranenburgh, S. Blending Computer Vision into Discrete Choice Models. Preprint. 2020. Available online: [https://transp-or.epfl.ch/heart/2020/abstracts/HEART\\_2020\\_paper\\_109.pdf](https://transp-or.epfl.ch/heart/2020/abstracts/HEART_2020_paper_109.pdf) (accessed on 5 May 2024).
2. Ben-Akiva, M.; Lerman, S. *Discrete Choice Analysis: Theory and Application to Travel Demand*; MIT Press Series in Transportation Studies; MIT Press: Cambridge, MA, USA, 1985; ISBN 9780262022170.
3. Hagenauer, J.; Helbich, M. A comparative study of machine learning classifiers for modeling travel mode choice. *Expert Syst. Appl.* **2017**, *78*, 273–282. [CrossRef]
4. Acuna-Agost, R.; Delahaye, T.; Lheritier, A.; Bocamazo, M. Airline itinerary choice modelling using machine learning. In Proceedings of the International Choice Modelling Conference, Cape Town, South Africa, 3–5 April 2017.
5. Guo, C.; Berkhahn, F. Entity embeddings of categorical variables. *arXiv* **2016**, arXiv:1604.06737.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
7. Otsuka, M.; Osogami, T. A deep choice model. In Proceedings of the AAAI, Phoenix, AZ, USA, 12–17 February 2016; pp. 850–856.
8. Brathwaite, T.; Vij, A.; Walker, J.L. Machine learning meets microeconomics: The case of decision trees and discrete choice. *arXiv* **2017**, arXiv:1711.04826.

9. Sajjad, I.; Nafisah, I.A.; Almazah, M.M.A.; Alamri, O.A.; Dar, J.G. A Symmetrical Analysis of Decision Making: Introducing the Gaussian Negative Binomial Mixture with a Latent Class Choice Model. *Symmetry* **2024**, *16*, 908. [[CrossRef](#)]
10. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the International Conference on Machine Learning, Washington, DC, USA, 28 June–2 July 2011.
11. Wang, Z.; Li, H.; Rajagopal, R. Urban2Vec: Incorporating Street View Imagery and POIs for Multi-Modal Urban Neighborhood Embedding. *arXiv* **2020**, arXiv:2001.11101. [[CrossRef](#)]
12. Sifringer, B.; Lurkin, V.; Alahi, A. Enhancing discrete choice models with representation learning. *Transp. Res. Part B Methodol.* **2020**, *140*, 236–261. [[CrossRef](#)]
13. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
14. Verwimp, L.; Pelemans, J.; Wambacq, P. Expanding n-gram training data for language models based on morpho-syntactic transformations. *Comput. Linguist. Neth. J.* **2015**, *5*, 49–64.
15. Han, Y.; Zegras, C.; Pereira, F.C.; Ben-Akiva, M. A neuralembedded choice model: Tastenet-mnl modeling taste heterogeneity with flexibility and interpretability. *arXiv* **2020**, arXiv:2002.00922. [[CrossRef](#)]
16. Alwosheel, A.; van Cranenburgh, S.; Chorus, C.G. Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis. *J. Choice Model.* **2018**, *28*, 167–182. [[CrossRef](#)]
17. Iranitalab, A.; Khattak, A. Comparison of four statistical and machine learning methods for crash severity prediction. *Accid. Anal. Prev.* **2017**, *108*, 27–36. [[CrossRef](#)] [[PubMed](#)]
18. Wang, Y. A new concept using LSTM Neural Networks for dynamic system identification. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5324–5329.
19. Van Cranenburgh, S.; Wang, S.; Vij, A.; Pereira, F.; Walker, J. Choice modelling in the age of machine learning—discussion paper. *J. Choice Model.* **2022**, *42*, 100340. [[CrossRef](#)]
20. Camacho-Collados, J.; Pilehvar, M.T. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Intell. Res.* **2018**, *63*, 743–788. [[CrossRef](#)]
21. Paredes, M.; Hemberg, E.; O'Reilly, U.-M.; Zegras, C. Machine learning or discrete choice models for car ownership demand estimation and prediction? In Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, Italy, 26–28 June 2017; pp. 780–785.
22. Foudeh, P.; Salim, N. An ontology-based, fully probabilistic, scalable method for human activity recognition. *arXiv* **2021**, arXiv:2109.02902.
23. Perone, C.S.; Silveira, R.; Paula, T.S. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv* **2018**, arXiv:1806.06259.
24. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*.
25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
26. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
27. Wong, M.; Farooq, B. Reslogit: A residual neural network logit model for data-driven choice modelling. *Transp. Res. Part C Emerg. Technol.* **2021**, *126*, 103050. [[CrossRef](#)]
28. Feng, S.; Cong, G.; An, B.; Chee, Y.M. Poi2vec: Geographical latent representation for predicting future visitors. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
29. De Brébisson, A.; Simon, É.; Auvoilat, A.; Vincent, P.; Bengio, Y. Artificial neural networks applied to taxi destination prediction. *arXiv* **2015**, arXiv:1508.00021.
30. Wang, Z.; Xiao, D.; Fang, F.; Govindan, R.; Pain, C.; Guo, Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int. J. Numer. Methods Fluids* **2018**, *86*, 255–268. [[CrossRef](#)]
31. Wang, B.; Shaaban, K.; Kim, I. Revealing the hidden features in traffic prediction via entity embedding. *Pers. Ubiquitous Comput.* **2021**, *25*, 21–31. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.