

## Article

# Exploiting Spiking Neural Networks for Click-Through Rate Prediction in Personalized Online Advertising Systems

Albin Uruqi  and Iosif Viktoratos \* 

Department of Computer Science, ACT College, Thessaloniki, Vasiliou Sevenidi 17, 555 35 Pilea, Greece; albinuruqi@gmail.com

\* Correspondence: ivictor@act.edu or viktorat@econ.auth.gr

## Abstract

This study explores the application of spiking neural networks (SNNs) for click-through rate (CTR) prediction in personalized online advertising systems, introducing a novel hybrid model, the Temporal Rate Spike with Attention Neural Network (TRA-SNN). By leveraging the biological plausibility and energy efficiency of SNNs, combined with attention-based mechanisms, the TRA-SNN model captures temporal dynamics and rate-based patterns to achieve performance comparable to state-of-the-art Artificial Neural Network (ANN)-based models, such as Deep & Cross Network v2 (DCN-V2) and FinalMLP. The models were trained and evaluated on the Avazu and Digix datasets, using standard metrics like AUC-ROC and accuracy. Through rigorous hyperparameter tuning and standardized preprocessing, this study ensures fair comparisons across models, highlighting SNNs' potential for scalable, sustainable deployment in resource-constrained environments like mobile devices and large-scale ad platforms. This work is the first to apply SNNs to CTR prediction, setting a new benchmark for energy-efficient predictive modeling and opening avenues for future research in hybrid SNN-ANN architectures across domains like finance, healthcare, and autonomous systems.

**Keywords:** artificial intelligence; machine learning; click-through rate (ctr); artificial neural networks; spiking neural networks



Academic Editor: Sonia Leva

Received: 3 June 2025

Revised: 14 July 2025

Accepted: 16 July 2025

Published: 18 July 2025

**Citation:** Uruqi, A.; Viktoratos, I. Exploiting Spiking Neural Networks for Click-Through Rate Prediction in Personalized Online Advertising Systems. *Forecasting* **2025**, *7*, 38. <https://doi.org/10.3390/forecast7030038>

**Copyright:** © 2025 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

Artificial intelligence (AI) is driving a rapid, unprecedented technological advancement, revolutionizing society and transforming human lifestyles [1]. The recent increase is attributed to neural networks (NNs) and the surge of deep learning (DL), one of AI's subfields. Neural networks (NNs) are capable of processing and interpreting large amounts of data, making them remarkably effective for complex tasks such as forecasting, image recognition, natural language processing, and artistic creation [2]. AI aims to extend and augment mankind's capabilities and efficiency in several distinct areas, such as business intelligence, health, transportation, and more. An area that has seen a phenomenal boost is business analytics. In the business world, AI has significantly boosted competitive advantage by offering optimization and automation tools that accelerate delivery and enable the development of innovative products and services. A key application of this technology lies in personalized online advertising, where predicting click-through rates (CTR)—the likelihood that a user will click on a specific ad—is fundamental to optimizing ad placement, maximizing user engagement, and increasing overall revenue. CTR prediction is a central component of real-time bidding and recommendation systems, where even small gains in

accuracy can translate into significant business impact. However, this task is highly challenging due to the massive scale of data, sparsity of user feedback, and the need to model complex, nonlinear interactions between user and ad features in real time [3]. However, AI systems also face challenges, including a lack of transparency in decision-making processes, potential biases embedded in training data, and high energy consumption, particularly for traditional deep learning models, which motivates the exploration of energy-efficient alternatives like spiking neural networks (SNNs) [4].

This study aims to advance personalized advertising systems by testing the efficacy of SNNs for CTR prediction, focusing on their accuracy and energy efficiency compared to traditional machine learning, ANNs, and state-of-the-art models [5]. By evaluating different SNN neuron classes, such as Leaky Integrate-and-Fire (LIF) and RSynaptic neurons, this research assesses their performance in capturing complex user behavior patterns. A novel SNN model is introduced, demonstrating accuracy comparable to state-of-the-art approaches while achieving significantly higher energy efficiency, a critical factor for sustainable deployment in resource-constrained environments like mobile devices and large-scale ad platforms. Through rigorous experimentation, this study highlights the potential of SNNs to redefine standards in CTR prediction. The energy efficiency of SNNs offers substantial benefits for reducing computational costs and environmental impact, making them a promising solution for next-generation personalized advertising algorithms.

This study begins with an introduction to personalized advertising systems and SNNs. It then provides an overview of current state-of-the-art approaches and outlines the contributions of this work, particularly the development of a novel energy-efficient SNN model with high accuracy. Subsequent sections detail the examined models, datasets, and experimental results, comparing accuracy across SNNs and traditional methods. Finally, the authors discuss the findings, highlighting the balance between accuracy and energy efficiency, conclude the work, and propose future research directions to further optimize SNNs for personalized advertising applications.

### 1.1. Personalized Online Advertising Systems

Personalized Online Advertising Systems, also known as targeted advertising, represent a significant paradigm shift in the world of digital marketing and commerce. These systems use algorithms heavily trained on huge amounts of user data to tailor the advertisements humans see on online platforms to match their unique tastes and needs [6]. User data contains many aspects of information, such as search queries, website visits, previous purchases, location, and demographic information about users. Accordingly, these systems aim to display ads precisely when the user is most likely to be interested in them [7]. According to Statista research, online advertising income in the US increased from USD 124.6 billion to USD 139.8 billion in 2020, a 12.2 per cent increase over 2019. By 2025, the industry for online advertising is anticipated to grow to USD 982.82 billion [8]. Click-based performance measurements, such as the number of clicks or click-through rate (CTR), are used in online advertising to measure how relevant advertisements are to the users [9]. CTR is the ratio of clicks to the number of times an advertisement is shown.

$$\text{CTR} = \frac{\text{clicks}}{\text{impressions}} \times 100(\%) \quad (1)$$

Improving CTR is considered an effective strategy for driving long-term growth in the online advertising ecosystem. An improved CTR can lead to more effective ad spending, higher conversion rates, and better resource allocation, translating into significant economic benefits. Thus, industry experts and researchers use statistical and machine learning models to predict the likelihood that users would click on a certain link or element within a digital

interface, such as a website, email, or advertisement. This problem is known as CTR prediction [10].

### 1.2. Spiking Neural Networks (SNNs)

Neural networks are computational machine learning models inspired by human brain dynamics. These models consist of artificial interconnected dots, replicating brain neurons, that process and direct information. Most current Artificial Neural Network-based models are based on highly simplified brain dynamics. They have been used as powerful computational tools to solve complex pattern recognition, function estimation, and classification problems. However, due to the need for more powerful and biologically realistic models, spiking neural network-based models have been developed, which comprise spiking neurons instead of other traditional artificial neurons [11]. The process of spiking signal transmission is quite sophisticated and differs from the traditional continuous values concept. The spikes (action potentials) travel along axons and activate synapses. These synapses release a neurotransmitter that quickly diffuses across the synaptic cleft to bind to receptors on the post-synaptic membrane, thereby transmitting the signal to the next neuron, the post-synaptic neuron.

In the post-synaptic neuron, these neurotransmitters affect the neuron's membrane potential. With each incoming spike, the membrane potential of the neuron rapidly increases before slowly declining (inset). Suppose several spikes arrive in a short period of time. In that case, the membrane potential crosses a threshold, and a spike is generated and fired down the axon, simultaneously resetting its potential [12]. SNNs offer advantages in processing real-time data. Their ability to handle sparse and event-driven data makes them well-suited for scenarios such as CTR prediction, where user actions and events are sporadic. SNNs' energy efficiency and potential for asynchronous processing also have practical benefits for large-scale implementations [13].

### 1.3. Challenge and Objectives

This research aims to tackle the challenge of the CTR prediction problem using advanced SNN-based models and outperform existing state-of-the-art approaches. The objectives of this work are the following:

- Design and develop novel SNN-based models tailored to the specific requirements and challenges of personalized online advertising systems, considering several complex factors such as user behavior patterns and information;
- Investigate and understand the potential advantages of SNN-based models over traditional ANN-based models in predicting CTR in the context of personalized online advertising;
- Conduct empirical experiments and comparative analysis to quantitatively assess the performance of the SNN-based models in terms of CTR prediction accuracy, efficiency, and adaptability in comparison to traditional ANN-based models;
- Provide valuable insights for boosting advertising effectiveness for businesses and users. By refining CTR predictions, businesses can optimize their ad placements and targeting strategies, leading to better campaign performance and increased ROI. On the other hand, users will see more relevant ads, improving their overall online experience and engagement.

## 2. Related Work and Contribution

### 2.1. Related Work

SNNs have recently begun to gain more attention in a variety of fields in the tech industry. Some of the most recent state-of-the-art applications of SNNs include the following:

- Detection of weather images. The transmission of weather information from a location at specific time intervals has an impact on the living conditions of the people who live there, either directly or indirectly. Toğaçar et al. (2020) in their study focus on whether it is possible to make weather predictions based on weather images using today's technology and computer systems [14]. They demonstrate that using SNNs in conjunction with deep learning models can achieve a successful result;
- Recognition of emotional states. Luo et. al. (2020) propose an innovative method for recognizing emotion states by combining spiking neural networks (SNNs) and electroencephalograph (EEG) techniques [15]. Two datasets are used to validate the proposed method. The results show that using the variance data processing technique and SNNs can classify emotional states more accurately. Overall, this work achieves a better performance than the benchmarking approaches and demonstrates the advantages of using SNNs for emotion state classifications;
- Event-based optical flow estimation. Kosta et al. (2023) focus on improving the task of optical flow estimation using SNNs with learnable Neuronal Dynamics [16]. Experiments on the Multi-Vehicle Stereo Event-Camera dataset and the DSEC-Flow dataset show a reduction in average endpoint error (AEE) compared to state-of-the-art ANNs.

Regarding the domain of CTR prediction, multiple efforts have been made in recent years to improve the accuracy. Some of the most recent approaches include the following:

- Combining convolutional neural networks (CNNs) and Factorization Machine (FM). Xiang-Yang et al. (2018) developed a model that extracts high-impact features with CNN and classifies them with FM, which can learn the relationship between feature components that are mutually distinct [17]. The experimental results show that the CNN-FM hybrid model can effectively improve the accuracy of advertising CTR prediction when compared to the single-structure model;
- Optimized Deep Belief Nets (DBNs). Chen et al. (2019) focus on increasing the training efficiency of a prediction model using DBNs by proposing a network optimization fusion model based on a stochastic gradient descent algorithm and an improved particle swarm optimization algorithm. The experiment results show that the fusion method improved the training efficiency of deep belief nets, resulting in a better CTR accuracy compared to the traditional models [18];
- Enhanced Parallel Deep CTR model. Current deep CTR models, such as Deep and Cross Networks (DCN), suffer from insufficient information sharing, which limits their effectiveness. Chen et al. (2021) propose an Enhanced Deep & Cross Network (EDCN) with lightweight, model-agnostic bridge and regulation modules, to enhance CTR prediction accuracy [19]. Extensive testing and real-world deployment on Huawei's online advertising platform show that EDCN outperforms the baseline models;
- Deep Field-Embedded Factorization Machines (Deep FEFMs). Pande et al. (2023) showed that a Deep FEFM-based model, compared to other Field Factorization Machine-based models, has much lower complexity [20]. Deep FEFM combines the interaction vectors learned by the FEFM component with a Deep NN and is, therefore, able to learn higher-order interactions. Also, after conducting comprehensive experiments on two publicly available real-world datasets, the results showed that Deep FEFM outperformed the existing state-of-the-art shallow and deep models;
- Ensemble multiple models to improve results. Viktoratos et al. (2022) proposed a novel approach to designing personalized advertisement systems by leveraging device and application-related variables from cold-start users [21]. This system trains and tests state-of-the-art machine learning models on existing datasets and uses a deep learning ensembler to predict CTR.

## 2.2. Contribution

The scientific and potential social contribution of this study can be encapsulated in the following key aspects:

- This research proposes a novel approach, which is the first to investigate the use of SNNs in the CTR prediction problem;
- The SNN-based models that have been developed and tested are highly competitive and more energy efficient than current state-of-the-art approaches in CTR prediction, which would have a significant impact on the online advertising industry;
- The introduced SNN-based models demonstrate a versatile approach that can be applied to various domains beyond CTR prediction, such as financial forecasting, healthcare diagnostics, autonomous systems, and natural language processing. This work contributes to neural network research by providing a framework for implementing SNNs in real-world applications with temporal dynamics and event-driven data processing, guiding researchers and practitioners in leveraging SNNs' unique characteristics like energy efficiency and temporal information processing across diverse problem domains;
- Potential economic and social impact for businesses and users. By improving the accuracy of ad targeting, businesses can increase revenue through higher click-through rates and conversions, while also providing a more personalized and relevant ad experience for users. This not only leads to economic benefits through improved resource allocation and increased sales but also enhances user satisfaction by reducing irrelevant ad exposure, ultimately fostering a more efficient and user-friendly digital ecosystem.

## 3. Materials and Methods

The research methodology included three phases:

1. Phase 1—First, well-known advertising datasets from publicly available sources were selected. Data preprocessing was conducted to ensure quality and consistency;
2. Phase 2—A baseline machine learning algorithm, a baseline artificial neural network (ANN), and a spiking neural network (SNN) were developed, along with more advanced state-of-the-art models, all specifically tailored for click-through rate (CTR) classification tasks;
3. Phase 3—In the third phase, the models were trained and tested on the datasets. All models were fine-tuned, exploring various neuron classes, learning rules, and network topologies to optimize performance for CTR prediction. Rigorous testing and validation were conducted using standard evaluation metrics, including AUC-ROC and accuracy. Throughout this process, iterative refinements were made based on experimental results to enhance the models' accuracy and efficiency. Finally, an extensive analysis of the results was performed to evaluate the effectiveness of SNN-based approaches for CTR prediction, highlighting their strengths and potential areas for future improvement in computational advertising.

### 3.1. Phase 1—Dataset Selection and Preprocessing

The training and testing phases of deep learning models are heavily dependent on data. Therefore, various online datasets were thoroughly examined. After evaluating key parameters such as size (rows and columns), data variety, and the number of variables, two datasets were selected for training and testing the developed models.

The first dataset has been released by Avazu Technology Company and is, therefore, usually referred to as the Avazu dataset. The Avazu dataset contains over 40 million rows and 24 columns, including 23 features and 1 target column. The dataset includes various

attributes related to online ad interactions. Each record is identified by a unique ad ID, with an outcome field indicating whether the ad was clicked or not. The dataset captures the timestamp of the interaction, formatted to show the exact hour, day, month, and year. It includes several anonymized categorical variables, along with information on the banner position and details about the associated site and app, such as their IDs, domains, and categories. The dataset also records device-specific information, including the device's ID, IP address, model, type, and connection type. Additionally, there are multiple anonymized categorical variables that further describe the data.

The second dataset is the Digix dataset, released in 2010. This dataset includes the advertising user behavior data collected from seven consecutive days. It has over 41 million rows and 36 columns. Similarly to the Avazu dataset, there is one target column representing the user click, and the rest of the columns represent the features. Features include fields that identify ad tasks, materials, and advertisers, as well as details about the ad's creative type, display form, and slot placement. User demographics such as age, city, occupation, and gender are captured, along with device-specific information like model, size, and network status during interactions. The dataset also classifies the apps involved in the ad tasks and provides information on app and device characteristics, such as storage size, release time, and ratings. Additionally, it includes data on user behavior, membership status, and activity levels, alongside details about the ad industry and the date of the recorded behavior. The two datasets utilized in this study are well-known and publicly available, having been extensively used in various academic competitions and research initiatives. These datasets have been anonymized to protect user identities and sensitive information, thereby mitigating privacy concerns.

To prepare the datasets for analysis, data preprocessing was performed on a subset of 0.15 million rows for both, selected to ensure an 80–20 distribution of 0s and 1s in the target variable ( $y$ ) in order to avoid significant class imbalance. Statistical analysis was used to eliminate variables that were uncorrelated with the target variable (user click). These data frames were split into training and validation (80%) and testing (20%) sets.

### 3.2. Phase 2—Models

To ensure fair and consistent comparisons across all models, we applied an identical preprocessing pipeline: categorical features were first tokenized into discrete indices and then mapped to dense vector representations via embedding layers. Numerical features were standardized through z-score normalization and linearly projected into a compatible vector space. All processed features—both embedded categorical and projected numerical—were then concatenated into a unified representation, flattened to a fixed dimension, and fed as input to each model. This standardized approach guarantees that performance differences reflect only architectural variations, not preprocessing artifacts. The following models were implemented for training and testing:

#### 3.2.1. Logistic Regression

Logistic regression is a baseline statistical machine learning technique for binary classification problems. The purpose of logistic regression models is to simulate the likelihood that an input falls into a particular class. The logistic function translates any real-valued integer to the range (0, 1).

#### 3.2.2. Deep Baseline Model

Deep Baseline models usually consist of multiple hidden layers. In this work, the Deep Baseline is built as a Sequential Keras model with four fully connected dense layers. The first layer is responsible for data input; then, the second and third layers are considered hidden layers, and the last layer is responsible for the output. The first three layers use the

rectifier function, which is known as the ReLU activation function. The last layer (output layer) comprising one neuron uses the sigmoid function. This model is compiled using the Adam optimizer.

### 3.2.3. RSynaptic Model

RSynaptic neuron class is one of the most popular neuron classes in SNNs. The network is made of four linear fully connected layers followed by RSynaptic neurons, which represent spiking neurons with synaptic plasticity. Each spiking neuron layer is defined with specific parameters, including the number of neurons, initial membrane potential ( $V$ ), learning thresholds ( $\alpha$  and  $\beta$ ), and reset mechanism.

The synaptic current jumps upon spike arrival, which causes a jump in membrane potential. Synaptic current and membrane potential decay exponentially with rates of  $\alpha$  and  $\beta$ , respectively. For

$$U[T] > U_{thr} \Rightarrow S[T + 1] = 1. \quad (2)$$

Whenever an RSynaptic neuron emits a spike,

$$\begin{aligned} I_{syn}[t + 1] &= \alpha I_{syn}[t] + VS_{out}[t] + I_{in}[t + 1] \\ U[t + 1] &= \beta U[t] + I_{syn}[t + 1] - R(\beta U[t] + I_{syn}[t + 1]) \end{aligned} \quad (3)$$

where

$I_{syn}$ —Synaptic current;  
 $I_{in}$ —Input current;  
 $U$ —Membrane potential;  
 $U_{(thr)}$ —Membrane threshold;  
 $S_{out}$ —Output spike;  
 $R$ —Reset mechanism;  
 $\alpha$ —Synaptic current decay rate;  
 $\beta$ —Membrane potential decay rate;  
 $V$ —Explicit recurrent weight.

In RSynaptic architecture, the forward method implements the forward pass of the network. During each time step, input signals ( $x$ ) are processed consequently through the layers. Spikes, synaptic weights, and membrane potentials are updated at each layer using the 'RSynaptic' neuron model.

### 3.2.4. Leaky Model

A Leaky Spiking Network is another form of spiking neural networks. It utilizes Leaky Integrate-and-Fire (LIF) neurons, a kind of spiking neuron model in which the dynamics of the membrane potential include a leakage factor. It is assumed that the input is an injection of current. By leveraging rate  $\beta$ , the membrane potential decays exponentially. For

$$U[T] > U_{thr} \Rightarrow S[T + 1] = 1. \quad (4)$$

Whenever a Leaky neuron emits a spike,

$$U[t + 1] = \beta U[t] + I_{syn}[t + 1] - R(\beta U[t] + I_{in}[t + 1]) \quad (5)$$

where

$I_{in}$ —Input current;  
 $U$ —Membrane potential;

U<sub>thr</sub>—Membrane threshold;  
 R—Reset mechanism;  
 $\beta$ —Membrane potential decay rate.

The Leaky neurons have parameters such as beta (leakiness), threshold, and reset mechanism. The forward pass initializes the membrane potentials for each Leaky neuron in every layer. It iterates over time steps and then updates the membrane potentials at each step. Only the output of the last layer is recorded for each time step. The final layer includes the spike trains, and the membrane potentials of the previous layer are extended along the time dimension.

### 3.2.5. RLeaky Model

An RLeaky Spiking Network is an extensive format of the previously mentioned Leaky network. Therefore, these two networks are remarkably similar in the way communication within the network happens. However, the main difference is that in the RLeaky network, the neurons utilize the Randomized Leaky function, which introduces randomness into the leakage process. It is assumed that the input is an injection of current appended to the voltage spike output. Similarly to the Leaky network, by leveraging rate beta, the membrane potential decays exponentially. However, compared to the Leaky network, there is a dynamic leak rate that varies randomly over time. For

$$U[T] > U_{thr} \Rightarrow S[T + 1] = 1. \quad (6)$$

Whenever an RLeaky neuron emits a spike,

$$U[t + 1] = \beta U[t] + I_{in}[t + 1] + V(S_{out}[t]) - R(\beta U[t] + I_{in}[t + 1] + V(S_{out}[t])) \quad (7)$$

where

I<sub>in</sub>—Input current;  
 U—Membrane potential;  
 U<sub>thr</sub>—Membrane threshold;  
 S<sub>out</sub>—Output spike;  
 R—Reset mechanism;  
 $\beta$ —Membrane potential decay rate;  
 V—Explicit recurrent weight.

The forward pass function is similar to the Leaky pass function, with the exception of the dynamic leak rate.

### 3.2.6. Deep & Cross Network v2 Model (DCN v2)

DCN-v2 (Deep & Cross Network v2) is an enhanced version of the original Deep & Cross Network, specifically designed for automatic learning of explicit feature interactions in click-through rate (CTR) prediction tasks [19]. The architecture offers two structural configurations: a parallel structure where cross and deep networks run independently with concatenated outputs, and a stacked structure where one network's output serves as input to the other. Its architecture components include the following:

- A. Cross-Layer—The cross-layer network represents the core innovation of DCN models. DCN-v2 significantly improves upon its predecessor by replacing the single weight vector of DCN-v1 with a weight matrix  $W$ , substantially increasing the model's expressiveness. This enhancement enables more sophisticated feature interaction modeling while maintaining the network's ability to learn explicit high-order feature crosses efficiently;

- B. Deep Layers—Standard fully connected with multiple hidden layers with ReLU activation. Applied to the same input embeddings as the cross-layer network. Learns implicit high-order feature interactions;
- C. Integration Layer—The final layer combines representations from both the cross and deep networks. For binary classification tasks like CTR prediction, this typically involves a single output neuron with sigmoid activation that produces the final prediction probability by leveraging the complementary strengths of both network components.

### 3.2.7. Final MLP Model

FinalMLP is an advanced dual-branch neural network for CTR prediction that combines feature selection, parallel processing, and sophisticated feature interaction mechanisms [22]. It is designed to capture both individual feature patterns and cross-feature interactions effectively. Its architecture components include the following:

- A. Feature Selection Module—Dynamically selects relevant features for each MLP branch using attention-like gating. Contains two independent gates, one for each MLP branch. Can use specific features as context for gating decisions. Sigmoid gates multiply feature embeddings element-wise.

$$\text{gate1} = \text{Sigmoid}\left(\text{MLP}_{\text{gate1}}(\text{context1}_{\text{embeddings}})\right) \times 2 \quad (8)$$

$$\text{gate2} = \text{Sigmoid}\left(\text{MLP}_{\text{gate2}}(\text{context2}_{\text{embeddings}})\right) \times 2 \quad (9)$$

Outputs two differently weighted feature representations;

- B. Dual MLP Branches—follow the same pattern, but can have different configurations.
  - MLP1: Focuses on high-capacity feature learning (more parameters);
  - MLP2: Focuses on refined, compact representations (fewer parameters);
  - Different Dropouts: Varying regularization strengths;
- C. Interaction Aggregation Module—the most sophisticated component—is a multihead bilinear interaction module. Mathematically,

$$\text{output} = W_x \times x + W_y \times y + \sum_{h=1}^H x_h^T \times W_h \times y_h \quad (10)$$

where

- $x, y$ : Outputs from MLP1 and MLP2;
- $W_x, W_y$ : Linear projection weights;
- $W_h$ : Bilinear interaction matrix for head  $h$ ;
- $H$ : Number of heads (default: 2);
- D. Output Layer—Converts logits to probabilities for CTR prediction. Output Range:  $[0, 1]$  represents click probability.

### 3.2.8. Temporal Rate Spike with Attention NN Model (TRA-SNN)

TRA-SNN is a novel architecture that combines traditional deep learning embeddings with spiking neural networks for click-through rate prediction (Figure 1). This model leverages biological plausibility and energy efficiency of spiking neurons while maintaining competitive performance through dual-branch processing and attention-based feature selection. Architecture components are as follows:

- A. Feature Selection Module—Similar to final MLPs, attention-based feature selection to identify important feature interactions. Multi-head self-attention with 2 attention heads captures inter-field dependencies and contextual relationships. Then, gating

with a sigmoid is used to create two feature representations from the flattened embeddings. In detail:

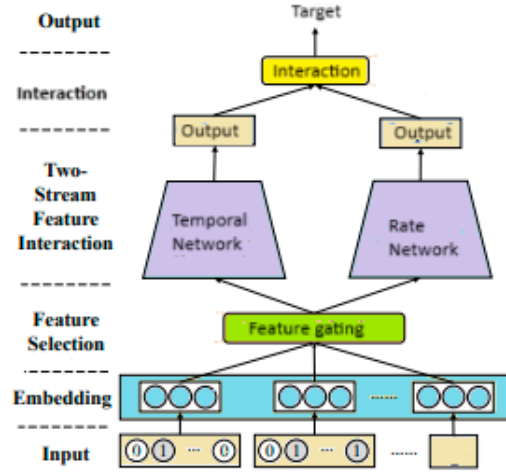


Figure 1. TRA-SNN architecture.

Let  $X \in \mathbb{R}^{b \times d}$  denote the input feature embeddings, where  $b$  is the batch size and  $d$  is the total embedding dimension (e.g.,  $d = 23 \times 32 = 736$  for DigiX with 22 categorical and 1 numeric feature, embedding size 32). The attention mechanism computes query (Q), key (K), and value (V) matrices for each head  $h \in \{1, 2\}$ :

$$Q_h = X \cdot W_{Q_h}, K_h = X \cdot W_{K_h}, V_h = X \cdot W_{V_h}, \text{ where } W_{Q_h}, W_{K_h}, W_{V_h} \in \mathbb{R}^{d \times d_h} \quad (11)$$

where  $d_h = d/2$  is the dimension per head (e.g.,  $d_h = 368$  for DigiX). The attention scores are computed as

$$A_h = \text{softmax}\left(Q_h \cdot K_h^T / \sqrt{d_h}\right), \text{ Attention}_h(X) = A_h \cdot V_h \quad (12)$$

The outputs from both heads are concatenated and passed through a linear layer to produce a unified representation:

$$Z = \text{Concat}(\text{Attention}_1(X), \text{Attention}_2(X)) \cdot W_Z + b_Z, \text{ where } W_Z \in \mathbb{R}^{d \times d}, b_Z \in \mathbb{R}^d \quad (13)$$

Two sigmoid gates generate weighted feature representations for the temporal and rate branches:

$$G_1 = \sigma(W_{G_1} \cdot Z + b_{G_1}), G_2 = \sigma(W_{G_2} \cdot Z + b_{G_2}) \quad (14)$$

$$X_{\text{temporal}} = G_1 \odot Z, X_{\text{rate}} = G_2 \odot Z \quad (15)$$

where  $\odot$  denotes element-wise multiplication, and  $\sigma$  is the sigmoid function. The outputs  $X_{\text{temporal}}$  and  $X_{\text{rate}}$  are fed to the respective SNN branches;

#### B. Temporal and Rate SNN Branches

**Temporal branch**—This branch uses RSynaptic neurons to capture temporal dynamics over multiple time steps. Each layer applies a linear transformation followed by RSynaptic neuron processing. For a layer  $l$  with input  $X^{(l-1)} \in \mathbb{R}^{b \times n_{l-1}}$ , the synaptic current  $I_{\text{syn}}^{(l)}[t]$  and membrane potential  $U^{(l)}[t]$  at time step  $t$  are updated as

$$I_{\text{syn}}^{(l)}[t] = \alpha I_{\text{syn}}^{(l)}[t-1] + W^{(l)} X^{(l-1)}[t] + V S_{\text{out}}^{(l)}[t-1] \quad (16)$$

$$U^{(l)}[t] = \beta U^{(l)}[t-1] + I_{\text{syn}}^{(l)}[t] \quad (17)$$

where  $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$  is the weight matrix;  $V \in \mathbb{R}^{n_l \times n_l}$  is the recurrent weight;  $\alpha$  0.9 and  $\beta$  are decay rates, and  $S_{\text{out}}^{(l)}[t-1] \in [0, 1]^{b \times n_l}$  is the spike output from the previous time step. A spike is emitted if the membrane potential exceeds the threshold  $U_{\text{thr}}$ :

$$S_{\text{out}}^{(l)}[t] = 1 \text{ if } U^{(l)}[t] \geq U_{\text{thr}}, \text{ else } 0 \quad (18)$$

After a spike, the membrane potential is reset using the ‘subtract’ mechanism for Avazu:  $U^{(l)}[t] \leftarrow U^{(l)}[t] - U_{\text{thr}}$  or no reset for DigiX. The final layer output  $S_{\text{temporal}}[T] \in \mathbb{R}^{b \times n_L}$  represents the temporal spike train.

Rate branch—This branch aggregates spike counts over time steps to produce a rate-based representation. For each layer  $l$ , the spike output  $S_{\text{out}}^{(l)}[t]$  is computed as above, and the average spike rate is calculated as

$$R^{(l)} = (1/T) \sum_{t=1}^T S_{\text{out}}^{(l)}[t] \quad (19)$$

The rate output  $R_{\text{rate}} \in \mathbb{R}^{b \times n_L}$  from the final layer is defined as

$$X_{\text{rate}} = R^{(l)} \quad (20)$$

The branch architecture mirrors the temporal branch for consistency.

### C. Interaction Aggregation Module

The multi-head factorized bilinear interaction module fuses the temporal and rate branch outputs ( $X_{\text{temporal}}, X_{\text{rate}} \in \mathbb{R}^{b \times n_L}$ ) to produce the final CTR prediction.

The outputs are projected into head-specific dimensions ( $H = 2$ ):

$$X_{\text{headh}} = X_{\text{temporal}} \cdot W_{x_h}, Y_{\text{headh}} = X_{\text{rate}} \cdot W_{y_h} \quad (21)$$

$$\text{where } W_{x_h}, W_{y_h} \in \mathbb{R}^{n_L \times d_h}, d_h = n_L / H \quad (22)$$

The bilinear interaction for each head  $h$  is computed as

$$Z_h = \sum \left( \left( X_{\text{headh}}^\top U_h \right) \odot \left( Y_{\text{headh}}^\top V_h \right) \right), \# \text{ shape : } (b \times 1) \quad (23)$$

where

- $b$  is the batch size and  $n_L$  the feature dimension;
- $W_{x_h}, W_{y_h} \in \mathbb{R}^{n_L \times d_h}$ ;
- $U_h, V_h \in \mathbb{R}^{d_h \times k}$ ;
- $\odot$  is element – wise multiplication;
- $k$  is the factorization dimension.

The head outputs are concatenated and passed through a linear layer to produce the final logits:

$$Z = \text{Concat}(Z_1, Z_2, \dots, Z_H) W_Z + b_Z, \quad (24)$$

$$y_{\text{pred}} = \sigma(Z) \quad (25)$$

This model is trained with a binary cross-entropy (BCE) loss for Avazu and a mean squared error (MSE) loss for DigiX.

$$\text{loss}_{\text{BCE}} = -(1/b) \sum_{i=1}^b \left[ y_{\text{true},i} \cdot \log(y_{\text{pred},i}) + (1 - y_{\text{true},i}) \cdot \log(1 - y_{\text{pred},i}) \right] \quad (26)$$

$$\text{loss}_{\text{MSE}} = (1/b) \sum_{i=1}^b (y_{\text{pred},i} - y_{\text{true},i})^2 \quad (27)$$

This module is crucial because it determines how information from parallel spiking pathways is integrated.

### 3.3. Phase 3—Training and Testing Models

Beginning with the training and the testing of ANN-based models, the Keras library was used for this purpose. Metrics such as Area Under the ROC Curve and Accuracy were calculated to test their performance. On the other hand, to train and test the SNN-based models, the snnTorch (an extension of Pytorch) library was utilized. Similarly, the same metrics, such as Area Under the ROC Curve and Accuracy, were calculated.

To provide a fair comparison between SNN and traditional ANN-based models in the predicting CTR problem, (a) the same feature extraction mechanism for embeddings was used, and (b) the most effective configuration parameters were identified for every model. This is performed through hyperparameter tuning techniques. Hyperparameter tuning is a critical aspect and a proven validation method for model development. Due to their importance in defining structural and operational characteristics, hyperparameters can significantly influence the models' performance [23]. This is performed by implementing various tuning techniques, starting with simpler ones such as manual trial-and-error, to more complicated tuning techniques such as grid search. Grid search is performed by manually selecting a subset of the hyperparameter space and testing [24,25]. Using the grid search algorithm, several optimal values of selected hyperparameters were found for all the models (see Table 1 for TRA-SNN). In order to find optimal values for other hyperparameters, manual trial and error have been utilized. It is worth mentioning that, for simple SNNs, the most effective technique was found to be the rate coding—counting the average spike rate over the time steps and using this spike rate to predict the target. For accuracy, if the rate was greater than or equal to 0.5, it was classified as 1. For AUC, the error was calculated between the spike rate and the prediction target.

**Table 1.** Optimal Values using Grid Search for TRA-SNN.

| Parameter                       | Digix                                       | Avazu                                       |
|---------------------------------|---------------------------------------------|---------------------------------------------|
| Branch Architecture             | 256–128–64                                  | 150–150–150                                 |
| Gradient Approximation Strategy | Sigmoid surrogate                           | Sigmoid surrogate                           |
| Learning Rate                   | 0.0001                                      | 0.0001                                      |
| Reset Mechanism (for SNN)       | None                                        | subtract                                    |
| Number of Time Steps (SNNs)     | 4                                           | 15                                          |
| Optimizer                       | Adam                                        | Adam                                        |
| Batch size                      | 2086                                        | 2086                                        |
| Embedding Size                  | 40                                          | 32                                          |
| Activation Functions            | Relu in hidden layers,<br>Sigmoid in output | Relu in hidden layers,<br>Sigmoid in output |
| Loss Function                   | BCE                                         | MSE                                         |
| Epochs                          | 10                                          | 10                                          |
| Dropout                         | 0.1                                         | 0.1                                         |
| Weight Initialization           | Xavier                                      | Xavier                                      |

## 4. Results

After data preprocessing and hyperparameter tuning, all models were trained and evaluated on both datasets. Starting with the Avazu dataset (see Table 2), RLeaky achieved the strongest performance among the spiking neuron models, while—as expected—the MLP outperformed all SNN variants. However, by leveraging both temporal and rate-based encoding mechanisms, the proposed TRA-SNN model delivered a notable performance

boost. TRA-SNN achieved an AUC of 0.7538, an accuracy of 0.809, and a log loss of 0.4331, performing on par with state-of-the-art deep learning models and slightly outperforming DCN-V2 (AUC 0.7531, accuracy 0.808, log loss 0.4334). In terms of performance improvement (PI) relative to TRA-SNN AUC, it significantly outperformed traditional machine learning models such as logistic regression lagged (PI = 15.0%), highlighting the considerable gap in predictive quality [19]. Furthermore, PI results were competitive with state-of-the-art models (e.g., with DCN-V2 0.09%, and with FinalMLP −0.19%), suggesting marginal differences (likely due to dataset-specific variance or minor overfitting effects). To evaluate further and formalize these findings, we conducted statistical significance testing using the DeLong test [26]. TRA-SNN demonstrated a statistically significant improvement in AUC over traditional machine learning methods such as logistic regression ( $p$ -value  $< 0.01$ ), validating the observed PI of over 15%. As expected, no statistically significant differences were found between TRA-SNN and other state-of-the-art deep models (e.g., DCN-V2, FinalMLP), which is reasonable given the small margins and overall saturation in performance. These results highlight TRA-SNN's robust balance of accuracy, generalization, and efficiency, offering a compelling alternative to conventional and deep models, particularly in energy-constrained or neuromorphic applications.

**Table 2.** Initial results using Avazu dataset.

| Model Name          | AUC ROC | Accuracy | Log Loss | PI(%) |
|---------------------|---------|----------|----------|-------|
| Logistic Regression | 0.6550  | 0.785    | 0.4743   | 15.0  |
| Leaky               | 0.7307  | 0.792    | 0.4502   | 3.16  |
| RLeaky              | 0.7411  | 0.796    | 0.4418   | 1.71  |
| RSynaptic           | 0.7487  | 0.798    | 0.4397   | 0.68  |
| Deep Baseline       | 0.7502  | 0.801    | 0.4356   | 0.48  |
| DCN-V2              | 0.7531  | 0.808    | 0.4334   | 0.09  |
| TRA-SNN             | 0.7538  | 0.809    | 0.4331   | -     |
| FinalMLP            | 0.7552  | 0.810    | 0.4319   | −0.19 |

For the Digix dataset (see Table 3), the proposed TRA-SNN model achieved an AUC of 0.7116, an accuracy of 0.8096, and a log loss of 0.4521, slightly outperforming DCN-V2 (AUC 0.7111, accuracy 0.8093, log loss 0.4523) and closely matching FinalMLP (AUC 0.7119, accuracy 0.8097, log loss 0.4512). Although the differences among top models were small, traditional methods like logistic regression lagged significantly, with an AUC of 0.6178, an accuracy of 0.8000, and a log loss of 0.4926. The TRA-SNN model demonstrated a performance improvement of 15.18%. Among spiking neuron models, RSynaptic performed strongly, with an AUC of 0.7073, an accuracy of 0.8061, and a log loss of 0.4585, highlighting the potential of spiking-based approaches to rival deep learning results. Other SNN variants, such as RLeaky and Leaky, showed promise but had slightly higher loss values. Once again, DeLong tests confirmed a statistically significant difference from logistic regression ( $p < 0.01$ ) and comparable performance with state-of-the-art approaches.

To evaluate the energy efficiency of each model, we also measured the number of million Multiply–Accumulate operations (MACs) required per inference—a standard proxy for computational cost in deep learning models. To ensure a fair comparison, all models were configured with the same embedding size (32) and 150 neurons in the hidden layers. As shown in Table 4, traditional deep models such as DCN-V2 and FinalMLP required 1.24–1.58 million and 0.42–0.51 million MACs, respectively, across the Avazu and Digix datasets. In contrast, TRA-SNN required significantly fewer MACs: only 0.31 million for Avazu and 0.39 million for Digix. This translates to an energy efficiency improvement (EEI) of 74.7–75.2% over DCN-V2, and 23.1–24.8% over FinalMLP. While these reductions already highlight the benefits of TRA-SNN under standard hardware assumptions, its true

energy-saving potential becomes even more evident on neuromorphic hardware platforms such as Intel’s Loihi or BrainScaleS, where spiking neural networks (SNNs) operate using event-driven accumulate-only operations (ACs) instead of MACs. These systems can achieve orders-of-magnitude lower energy consumption for SNNs due to their sparse and asynchronous nature, which traditional models cannot leverage. Thus, the observed improvements in MAC-based energy efficiency likely underestimate the full advantage of TRA-SNN when deployed on neuromorphic edge devices.

**Table 3.** Results using Digix dataset.

| Model Name          | AUC ROC | Accuracy | Log Loss | PI(%) |
|---------------------|---------|----------|----------|-------|
| Logistic Regression | 0.6178  | 0.8000   | 0.4926   | 15.18 |
| Leaky               | 0.6906  | 0.8038   | 0.4694   | 3.04  |
| RLeaky              | 0.6944  | 0.8045   | 0.4613   | 2.48  |
| RSynaptic           | 0.7073  | 0.8061   | 0.4585   | 0.61  |
| Deep Baseline       | 0.7098  | 0.8075   | 0.4547   | 0.25  |
| DCN-V2              | 0.7111  | 0.8093   | 0.4523   | 0.07  |
| TRA-SNN             | 0.7116  | 0.8096   | 0.4521   | -     |
| FinalMLP            | 0.7119  | 0.8097   | 0.4512   | −0.04 |

**Table 4.** Energy efficiency results.

| Model Name | MACs Avazu | MACs Digix | Average EEI(%) |
|------------|------------|------------|----------------|
| DCN-V2     | 1.2417     | 1.5826     | 75%            |
| FinalMLP   | 0.4180     | 0.5099     | 24%            |
| TRA-SNN    | 0.3142     | 0.3924     | -              |

Over the past few years, traditional artificial neural network (ANN)-based models—especially those incorporating attention mechanisms—have consistently outperformed spiking neural networks (SNNs) in most real-world tasks. The dominance of ANN architectures such as Transformers and deep MLPs is largely due to their continuous differentiability, ease of optimization, and superior expressiveness when dealing with complex data patterns. In contrast, the discrete nature of spike-based communication in SNNs presents significant challenges for gradient-based learning. Specifically, the reliance on surrogate gradients to train spiking neurons often results in less stable convergence and reduced overall model performance, limiting the broader adoption of SNNs in high-performance machine learning pipelines.

However, the results of this research challenge this prevailing narrative by showing that SNNs, when carefully designed and integrated with ANN components and attention mechanisms, can match the performance of state-of-the-art deep learning models. The proposed TRA-SNN model effectively captures both the temporal dynamics and rate-based coding properties of spiking neurons, while benefiting from the structured representations provided by attention modules. This hybrid approach addresses some of the core limitations of pure SNN models, particularly in learning efficiency and representational power. Most notably, the TRA-SNN model achieves performance comparable to deep learning baselines like DCN-V2 and FinalMLP on the Avazu dataset, as demonstrated in Table 2. These findings highlight a promising direction: leveraging the energy efficiency and temporal precision of SNNs in combination with the learning capacity of ANNs. This synergy opens up pathways for building neuromorphic models that are not only competitive in terms of accuracy but also highly efficient in terms of computational and energy resources. Such models are particularly relevant for deployment in edge computing scenarios, wearable devices, or any application where low-power, real-time inference is critical.

In summary, while pure SNNs may still face hurdles in standalone performance, this research demonstrates that hybrid SNN-ANN architectures enriched with attention mechanisms offer a viable and energy-efficient alternative to fully ANN-based models, without compromising accuracy.

## 5. Conclusions

This work explores the potential of SNNs for predicting CTR in personalized online advertising systems. Following a comprehensive examination of various spiking neuron classes and techniques, an innovative hybrid spiking model called TRA-SNN was developed. This model was systematically developed, trained, and evaluated using the Avazu and DigiX datasets, with AUC, accuracy, log loss, and MACs serving as the primary performance metrics for comparison against well-established methods. Empirically, TRA-SNN achieved competitive or superior performance across both benchmark CTR datasets, attaining AUC scores of 0.7538 and 0.7116 on Avazu and DigiX, respectively, with accuracies exceeding 0.809. When benchmarked against state-of-the-art deep learning models, including DCN-V2, FinalMLP, and TRA-SNN, it demonstrated comparable predictive capabilities while offering significant computational advantages. Our detailed analysis of computational cost through MAC operations revealed substantial efficiency gains—TRA-SNN achieved up to 75% reduction in MACs compared to DCN-V2 and 24% reduction over FinalMLP under controlled, fair experimental conditions. These results highlight TRA-SNN's inherent efficiency even under standard hardware assumptions, with the potential for even greater energy advantages on specialized neuromorphic hardware where spike-based architectures can exploit event-driven, accumulate-only computations for orders-of-magnitude reductions in energy consumption.

Altogether, these results position TRA-SNN as a statistically validated, empirically robust, and practically valuable solution for energy-constrained environments such as mobile devices, edge computing platforms, and real-time advertising systems. This research contributes significantly to the computational advertising domain. First, TRA-SNN introduces a novel hybrid spiking architecture specifically designed for CTR prediction, bridging biological plausibility with practical performance requirements. Second, this work provides the first comprehensive SNN evaluation for CTR prediction, establishing baseline metrics and demonstrating viability for real-world applications. Third, this rigorous computational analysis framework quantifies energy efficiency advantages through MAC operation counting, offering insights for deployment on conventional and neuromorphic platforms. Finally, this research advances theoretical understanding of temporal spike-based processing for static tabular data, opening avenues for energy-efficient machine learning in resource-constrained environments.

TRA-SNN presents a promising approach for CTR prediction, though certain aspects warrant further study to fully realize its potential. This model's performance relies on careful configuration of temporal parameters such as time steps, decay rates, and reset mechanisms, suggesting that additional research into adaptive tuning methods could enhance its robustness across diverse datasets. While TRA-SNN efficiently handles rich feature representations, its behavior in scenarios with high sparsity or cold-start conditions remains an area for exploration, as the temporal dynamics of spiking neurons may require specialized techniques to optimize signal accumulation. The inclusion of attention and bilinear interactions, while beneficial for modeling precision, invites future work on architectural refinements to ensure scalability in extremely large-scale applications. Although the model's hybrid design offers unique advantages, the interpretability of its spiking mechanisms compared to conventional deep learning approaches could benefit from further visualization and analysis tools to improve stakeholder understanding. Finally, while TRA-

SNN demonstrates notable efficiency gains in theory, empirical validation across different hardware platforms—including neuromorphic and conventional systems—would help clarify its practical energy-saving potential and guide implementation strategies. These considerations highlight opportunities for continued refinement rather than fundamental limitations, positioning TRA-SNN as a compelling direction for future research in efficient and biologically inspired recommendation systems.

The findings open promising directions for future research, including further refinement of the hybrid TRA-SNN architecture, integration of more sophisticated attention mechanisms, and enhanced spike-response mechanisms. The successful development of TRA-SNN also suggests potential for similar hybrid approaches in other domains beyond advertising, including finance, healthcare, and autonomous systems, positioning these innovative architectures as compelling alternatives to traditional deep learning models for tasks requiring both high performance and low energy consumption [27]. Overall, this work advances the understanding of SNNs through the development of the ground-breaking TRA-SNN model, demonstrating that hybrid spiking architectures can serve as both high-performing and energy-efficient tools that match state-of-the-art performance in CTR prediction while opening new possibilities for sustainable AI applications across diverse domains.

**Author Contributions:** Conceptualization, I.V.; methodology, I.V. and A.U.; software, A.U. and I.V.; validation, A.U. and I.V.; formal analysis, A.U.; investigation, I.V.; resources, A.U. and I.V.; data curation, A.U. and I.V.; writing—original draft preparation, A.U.; writing—review and editing, I.V.; visualization, A.U.; supervision, I.V.; project administration, I.V.; funding acquisition, I.V. All authors have read and agreed to the published version of the manuscript.

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

**Institutional Review Board Statement:** Not Applicable.

**Informed Consent Statement:** Not Applicable.

**Data Availability Statement:** Data are contained within the article.

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

## References

1. Liu, J.; Kong, X.; Xia, F.; Bai, X.; Wang, L.; Qing, Q.; Lee, I. Artificial Intelligence in the 21st Century. *IEEE Access* **2018**, *6*, 34403–34421. [CrossRef]
2. Neural Networks and Deep Learning. Available online: [https://www.ijareeie.com/upload/2025/february/27\\_Neural.pdf](https://www.ijareeie.com/upload/2025/february/27_Neural.pdf) (accessed on 10 October 2023).
3. de Medeiros, M.M.; Hoppen, N.; Maçada, A.C.G. Data science for business: Benefits, challenges and opportunities. *Bottom Line* **2020**, *33*, 149–163. [CrossRef]
4. Huang, P.; Gong, B.; Chen, K.; Wang, C. Energy-Efficient Neural Network Acceleration Using Most Significant Bit-Guided Approximate Multiplier. *Electronics* **2024**, *13*, 3034. [CrossRef]
5. Stanojevic, A.; Woźniak, S.; Bellec, G.; Cherubini, G.; Pantazi, A.; Gerstner, W. High-performance deep spiking neural networks with 0.3 spikes per neuron. *Nat. Commun.* **2024**, *15*, 6793. [CrossRef] [PubMed]
6. Sakalauskas, V.; Kriksciuniene, D. Personalized Advertising in E-Commerce: Using Clickstream Data to Target High-Value Customers. *Algorithms* **2024**, *17*, 27. [CrossRef]
7. Deng, S.; Tan, C.-W.; Wang, W.; Pan, Y. Smart Generation System of Personalized Advertising Copy and Its Application to Advertising Practice and Research. *J. Advert.* **2019**, *48*, 356–365. [CrossRef]
8. Research and Market Ltd. Online Advertising Market—Growth, Trends, COVID-19 Impact, and Forecasts (2021–2026). 2021. Available online: <https://www.researchandmarkets.com/reports/4602258/online-advertising-market-growth-trends#product-}-}-toc> (accessed on 10 October 2023).
9. Yang, Y.; Zhai, P. Click-through rate prediction in online advertising: A literature review. *Inf. Process. Manag.* **2022**, *59*, 102853. [CrossRef]

10. Viktoratos, I.; Tsadiras, A. Personalized Advertising Computational Techniques: A Systematic Literature Review, Findings, and a Design Framework. *Information* **2021**, *12*, 480. [\[CrossRef\]](#)
11. Islam, R.; Majurski, P.; Kwon, J.; Sharma, A.; Tummala, S.R.S.K. Benchmarking Artificial Neural Network Architectures for High-Performance Spiking Neural Networks. *Sensors* **2024**, *24*, 1329. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Varshika, M.L.; Corradi, F.; Das, A. Nonvolatile Memories in Spiking Neural Network Architectures: Current and Emerging Trends. *Electronics* **2022**, *11*, 1610. [\[CrossRef\]](#)
13. Ponulak, F.; Kasinski, A. Introduction to spiking neural networks: Information processing, learning and applications. *Acta Neurobiol. Exp.* **2011**, *71*, 409–433. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Toğaçar, M.; Ergen, B.; Cömert, Z. Detection of weather images by using spiking neural networks of deep learning models. *Neural Comput. Appl.* **2020**, *33*, 6147–6159. [\[CrossRef\]](#)
15. Luo, Y.; Fu, Q.; Xie, J.; Qin, Y.; Wu, G.; Liu, J.; Jiang, F.; Cao, Y.; Ding, X. EEG-Based Emotion Classification Using Spiking Neural Networks. *IEEE Access* **2020**, *8*, 46007–46016. [\[CrossRef\]](#)
16. Kosta, A.K.; Roy, K. Adaptive-SpikeNet: Event-based Optical Flow Estimation using Spiking Neural Networks with Learnable Neuronal Dynamics. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023. [\[CrossRef\]](#)
17. She, X.; Wang, S. Research on Advertising Click-Through Rate Prediction Based on CNN-FM Hybrid Model. In Proceedings of the 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 25–26 August 2018; pp. 56–59. [\[CrossRef\]](#)
18. Chen, J.; Zhang, Q.; Wang, S.; Shi, J.; Zhao, Z. Click-through Rate Prediction Based on Deep Belief Nets and Its Optimization. *J. Softw.* **2019**, *30*, 3665–3682. [\[CrossRef\]](#)
19. Chen, B.; Wang, Y.; Liu, Z.; Tang, R.; Guo, W.; Zheng, H.; Yao, W.; Zhang, M.; He, X. Enhancing Explicit and Implicit Feature Interactions via Information Sharing for Parallel Deep CTR Models. In Proceedings of the CIKM '21: 30th ACM International Conference on Information & Knowledge Management, New York, NY, USA, 1–5 November 2021. [\[CrossRef\]](#)
20. Pande, H. Field-Embedded Factorization Machines for Click-through rate prediction. *arXiv* **2021**. [\[CrossRef\]](#)
21. Viktoratos, I.; Tsadiras, A. A Machine Learning Approach for Solving the Frozen User Cold-Start Problem in Personalized Mobile Advertising Systems. *Algorithms* **2022**, *15*, 72. [\[CrossRef\]](#)
22. Mao, K.; Zhu, J.; Su, L.; Cai, G.; Li, Y.; Dong, Z. FinalMLP: An Enhanced Two-Stream MLP Model for CTR Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Washington, DC, USA, 7–14 February 2023.
23. Zhang, W.; Mi, X. A study on the efficiency and accuracy of neural network model to optimize personalized recommendation of teaching content. *Appl. Math. Nonlinear Sci.* **2025**, *10*, 1–23. [\[CrossRef\]](#)
24. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316. [\[CrossRef\]](#)
25. Weerts, H.J.P.; Mueller, A.C.; Vanschoren, J. Importance of Tuning Hyperparameters of Machine Learning Algorithms. *arXiv* **2020**. [\[CrossRef\]](#)
26. Zhu, H.; Liu, S.; Xu, W.; Dai, J. Linearithmic and unbiased implementation of DeLong's algorithm for comparing the areas under correlated ROC curves. *Expert Syst. Appl.* **2024**, *246*, 123194. [\[CrossRef\]](#)
27. Muir, D.R.; Sheik, S. The road to commercial success for neuromorphic technologies. *Nat. Commun.* **2025**, *16*, 3586. [\[CrossRef\]](#) [\[PubMed\]](#)

**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.