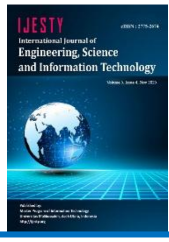
International Journal of Engineering, Science and Information Technology
Volume 5, No. 4 (2025) pp. 287-296
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i4.1242
Research Paper, Short Communication, Review, Technical Paper



An Efficient Hybrid Model-Based Classification of Online Personalized Ad and User Intent Detection Using CNN and Deep Q-Networks

Satish Babu Thunuguntla[1]*, S. Murugaanandam[2], Pitchai. R[3]

[1]Research Scholar, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
[2]Department of Networking and Communications, SOC, SRM Institute of Science & Technology, Kattankulathur, Tamil Nadu, India
[3]B V Raju Institute of Technology, Narsapur, Telangana, India

**Corresponding Author Email: st9189@srmist.edu.in*

The manuscript was received on 22 February 2025, revised on 15 May 2025, and accepted on 10 August 2025, date of publication 11 November 2025

Abstract

As a vast market, online advertising has promptly gained aggregate interest in a vast array of platforms, including mobile apps, search engines, third-party websites, and social media. In online marketing, the success of online campaigns is a challenge, which is often assessed by user response using several measures, such as product subscriptions, explicit customer feedback, clicks on ad creatives, or transactions obtained through online surveys. Auditing advertising images, or "creatives," prior to their appearance on publishers' websites is one of the most crucial quality control steps in online advertising. This ensures that advertisements only appear on websites that are appropriate for them. The user experience, the publisher's reputation, and maybe legal ramifications can all be negatively impacted if a sensitive creative is shown on the incorrect website. To detect and classify whether an advertisement has any sensitive content, we use a machine learning algorithm to process the creative image and combine this with the historical distribution of sensitive categories linked to the creative's landing page. To protect against this, the study presents an efficient hybrid model using CNN and Deep Q-networks for the classification of online personalized ads and user intent detection. Initially, the HCOAUICNN-DQN model uses data preprocessing to preprocess the input dataset from the online advertisement website. Next, extract those input features through the pretrained CNN-feature maps. Following, DQN is applied for the user intent detection classification of and online personalized ads. Finally, the hyperparameter tuning using Optuna optimization algorithm is applied to improve the real-time classification performance. A set of experiments was conducted to analyze model efficiency using different datasets of advertising images. The performance of the HCOAUICNN-DQN model will be assessed through accuracy, F1-score, recall, and precision metrics. The HCOAUICNN-DQN method obtains superior outcomes when compared to other existing approaches.

Keywords: *Online advertising, Convolutional Neural Network, Deep Neural Network, Optuna Optimization Algorithm, Machine Learning.*

1. Introduction

In modern economies, online marketing or online advertising is becoming more and more important [1]. With significant implications for the rest of society, online advertising is not only a large and quickly expanding business but also a major source of income for many publishers and suppliers of digital content, including blogs, newspapers, and review websites. Additionally, the prices that consumers pay for goods and services are influenced by advertising. It is crucial to understand the operation of this market for all of these reasons. Since the Internet can be leveraged to reach customers with marketing and promotional messages. Assigning the appropriate advertisements to the customers is the aim of online advertising in order to optimize the campaign's revenue, return on investment (ROI), or click-through rate (CTR). Real-time bidding (RTB) and guaranteed delivery (GD) are the two primary marketing techniques used in online advertising [2]. Ad exposures to consumers are ensured by contracts that are agreed beforehand between publishers and advertisers. Advertisers bid on each ad impression in real-time as it is created by a customer visit via real-time bidding. Nonetheless, most of the online advertising methods are based on static or offline optimization algorithms, which manage each impression separately and maximize the instantaneous revenue for each impression [3]. In real-world business, this can be a challenging problem, particularly when the environment is unstable. Since (i) showing inappropriate ads will degrade the advertising income of the platforms, and (ii) showing excessive ads or improper ads will diminish user engagement and experience.

In online advertising, an important quality control step is to audit advertising images (creatives) before they appear on publishers' websites [4]. In order to assure ad quality and a safe and minimally invasive experience for the user or users viewing the ad, this process makes sure that an ad is free of malware, follows an appropriate ad format, renders correctly each time, and does a number of other tests. Assigning different sensitive categories to advertisements is one of the most crucial steps in the creative audit process [5]. Although each ad exchange has its own guidelines and criteria, sensitive categories—also known as restricted—generally include items that may be suitable for some audiences but not for others, such as lingerie, alcohol, and tobacco [6]. One of the most crucial aspects of an audit is assigning the proper sensitive category to each creative. This classification safeguards brand safety, gives publishers complete control over the content on their websites, and keeps offensive information like hate speech and malvertising from ever showing up [7]. It might damage the publisher's reputation, harm the online user's experience, and have legal repercussions if a sensitive creative is shown on an improper website (e.g., targeting kids with alcohol advertising or other age banned content).

As a result [8], machine learning (ML), especially deep learning (DL), based approaches, are becoming increasingly popular for the user response prediction and online advertising classification, mainly because these approaches can simultaneously detect and classify whether an advertisement has any sensitive content [9], we use a ML algorithm to process the creative image and combine this with the historical distribution of sensitive categories linked to with the creative's landing page (the page that loads when the ad is clicked, which may also contain sensitive content). Every creative is checked for sensitive categories before it is shown on the website, which is costly and delays the creative from being displayed on our platform, while the majority of the audit procedure is at least partially automated. Furthermore, 95% of the creatives on our platform do not contain a sensitive category, suggesting that most of the automation opportunity occurs from reliably identifying non-sensitive ads [10].

The study presents an efficient HCOAUICNN-DQN technique for the classification of online personalized ads and user intent detection. Initially, the HCOAUICNN-DQN model applies data pre-processing to the raw input dataset obtained from the online advertisement website. Next, extract those input features through the pretrained CNN-feature maps. Following, DQN model for the classification and detection of online personalized ads and user preferences. Finally, the hyperparameter tuning using Optuna optimization algorithm is done for improving the real-time classification performance. A set of experiments was conducted to analyze the model's efficiency using different datasets of advertising images. The performance of the HCOAUICNN-DQN model will be assessed through accuracy, F1-score, recall, and precision metrics. The HCOAUICNN-DQN method obtains superior outcomes when compared to other existing approaches.

2. Literature Review

Jain et al. [11] compare and analyze several popular image classification approaches in order to identify the optimal method for the task of advertising. The most suitable approach for advertisement image classification is Convolutional Neural Networks (CNN), which automatically extracts features without requiring any previous knowledge of the features. This study further investigates and applies three dissimilar classification methods in order to classify ad images from online English newspapers into 4 pre-defined classifications involving job advertisements, admission announcements, tenders and sales, and promotional ads. These algorithms are trained and tested using a set of advertisement images gathered from four dissimilar online English newspapers. With results showing about 74% accuracy, the ResNet50 model that has been fine-tuned using "transfer learning" is determined to be the most appropriate method for these classification tasks. Newspaper readers will be able to conduct thorough advertisement searches in a category of their own interest with the aid of the presented method, saving them the time and effort of sequential human search through a range. This model will assist newspaper readers in performing comprehensive advertisement searches in a category of user interest, saving the time and effort needed for a sequential human search across a variety of newspapers.

Beauvisage et al. [12] examine this change in online advertising empirically. We demonstrate how three criteria—explainability, efficiency, and communicability —are used to evaluate targeted categories. The lasting role of demographic categories, the creation of audience segments tailored to individual advertisers, and the challenge of generalizing interest categories related to big data are all explained by the relative significance of these objectives. These findings highlight the prominence of examining the effects of cutting-edge AI and big data technologies in their respective professional and organizational settings of appropriation, as well as the need to consider the durability of the classifications that provide meaning to the social world.

Polina and Malathi [13] proposed a new CNN based image classification method for online advertising target customers, evaluating its accuracy against the extreme learning machine CNN (ELM-CNN) model. ELM-CNN with sample size of 10 and Wide Resnet with sample size 10 are used to determine the age and gender. Different ads will be shown based on the gender and age that have been identified. The Wide Resnet approach obtained a prediction accuracy of 93.8%, whereas the ELM-CNN method attained an accuracy of 82.6%. The statistical analysis yielded confidence values of 0.02 ($p < 0.05$). It demonstrates that Wide ResNet performs significantly better than the ELM-CNN model at identifying age and gender.

[14] Introduced a Session-aware GNN-based Recommendation, or SRA-GNN-Rec model for e-commerce, advertising, and marketing-based recommendation systems. Initially, SR-GNN-based model is used where the user preference is unknown and only access to current session is available. Next, the A-PGNN is applied where the user identity is known and one can leverage previous session information. The simulation outcomes on two publicly available datasets show that the SRA-NN-Rec performs reasonably well than other baseline models and is an effective mechanism for recommendations.

Joy and Deepthi, [15] suggested an ensemble architecture of ML and DL techniques to identify click fraud in online advertisement campaigns. The suggested model includes a CNN, a Random Forest (RF) for classification, and a Bidirectional LSTM (BiLSTM) for extracting hidden features. In order to automatically extract features from click data and process it through an RF classifier into two groups, including fraudulent and non-fraudulent clicks, the proposed model creates a hybrid DL model. To optimize the consistency and reliability of the clicks data, a preprocessing module is also created to cope with imbalanced data and categorical attributes. Additionally, the performance of the suggested CNN-Bi-LSTM-RF is assessed and compared with that of the ensemble and standalone models using several assessment criteria. Our ensemble design obtained 99.19 ± 0.08% accuracy, 99.89 ± 0.03% precision, 98.50 ± 0.11% sensitivity, 99.19 ± 0.08% F1-score, and 99.89 ± 0.03% specificity, according to the experimental data. In addition, the suggested model outperformed other ensembles and traditional models in terms of results. Additionally, our suggested ensemble architecture can be utilized to protect pay-per-click advertising from click fraud, enabling enterprises to promote their products securely and reliably.

[16] Proposed the attention-based user behavior modeling framework, ATRank and is primarily used for recommendation tasks. Our model takes into account heterogeneous user behaviors by projecting all behavior types into several latent semantic regions where self-attention might impact the actions. The user behavior vectors can then be used by downstream apps through vanilla attention. Tests demonstrate that ATRank can improve performance and speed up the training process. We also investigate ATRank's ability to forecast many user behavior types simultaneously using a single, cohesive model, demonstrating performance that is equivalent to that of the highly optimized individual models.

3. Methods

In this work, we present an efficient HCOAUICNN-DQN model for the classification of online personalized ads and user intent detection. Initially, the HCOAUICNN-DQN model applies data pre-processing to process the raw input dataset obtained from the online advertisement website. Next, extract those input features through the pretrained CNN-feature maps. Following is, DQN model for the classification and detection of online personalized ads and user preferences. Finally, the hyperparameter tuning using Optuna optimization algorithm is performed for improving the real-time classification performance. Figure 1 shows the overall architecture of the HCOAUICNN-DQN model.

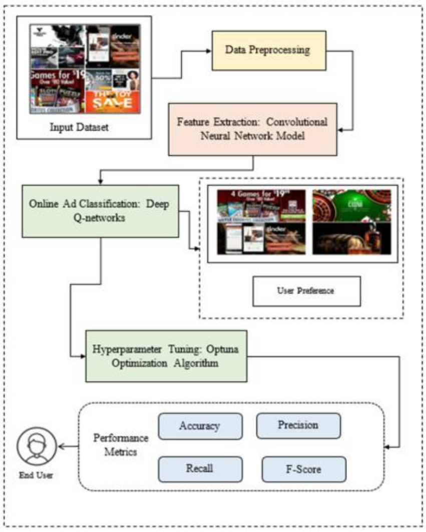


Fig 1. Overall Architecture of HCOAUICNN-DQN Model

3.1. Data Preprocessing

At first, the HCOAUICNN-DQN model applies data pre-processing to the raw input dataset obtained from the online advertisement website [17]. These advertisements, which include unstructured, inconsistent, and noisy content, frequently originate from a variety of sources. Particularly sensitive advertisements could use certain words, tones, or patterns that need to be properly maintained and recognized for better user preferences. In order to clean, normalize, and convert the raw textual dataset into a structured format appropriate for ML approaches, this technique eliminates unwanted and irrelevant content from the websites. In this study, we comprehensively examine how various pre-processing strategies described below leverage the ad classification task.

Data Cleaning

Eliminating HTML tags is essential for removing important text from web-scraped data or content from HTML sources. This phase ensures that relevant content remains to be studied.

Tokenization

This method uses character sequences with a white space between them to create tokens. When it comes to pre-processing tasks, this is the simplest tokenization technique. Typically, tokens are equivalent to a word, which splits ad textual data into tokens. Tokens could, however, be words combined with punctuation or other characters because the words have not undergone any preprocessing.

Words - Preprocessed

Before tokenization, sentences or documents are often preprocessed. We employ the following preprocessing techniques in our method, some of which are specific to the ad classification field:

a. Remove other non-alphanumeric characters.
b. Change ‘‘n’t’’ to ‘‘not.’’
c. Lowercase the sentence.
d. Eliminate ‘‘@name’’.
e. Isolate and eliminate punctuations except ‘‘?’’.

Remove Stop Words

Every sentence in the text dataset contains stop words, which add no meaning to the classification task. They are meaningless. Therefore, there is a need for a specific list of stop words for a single project. It is necessary to change the list of stop words and the classification of text data relevant to a particular language. Examples of stop words in the English language include myself, my, for, i, their, of, some, his, him, that, such, not no, can, re, will, ve, nor, and so on. Researchers do not exclude specific negative or positive terms from sentiment analysis algorithms because they help characterize the sentiments or reviews of the user. For example, Not, Less, No, Like, and so forth.

Stem

Stemming is a rule-based procedure that extracts a word's stem, or root form, by removing its suffixes. This method involves using word stems as tokens in classification analysis models. We employ the morphological parser for stemming [18]. Additionally, we employed a morphological parser to extract morphemes and lemmatize the text and image.

Lemma

Similar to stemming, Lemmatization extracts the root form of the words. The only difference is that irregular cases and morphophonemic rules are also taken into account. We employ the lemma forms of the words in the models, much like in stemming. Stanza is what we utilize for English lemmatization.

3.2. CNN Feature Extractor

Next, the input dataset was extracted through the pretrained CNN feature. Convolutional, pooling, and fully connected (FC) layers make up the hidden layer of CNNs, which also includes input, hidden, and output layers [19]. Convolutional layers of CNNs, as opposed to other neural networks, employ cross-covariance to only filter the input data from the lower layer while keeping their shape. This allows for the retention of both higher-level semantic features in the deeper layers and lower-level spatial features at the first layer close to the input layer. Additionally, CNNs can perform better in image processing than traditional NNs since spatial information can be preserved during the training process. The convolutional layers' filter or kernel serves as a container and produces a feature map from the input dataset. By minimizing the size of the data and enabling the features to have noticeable invariance across locations, the pooling layers speed up the training process. Similar to traditional multilayer perceptron networks, the final FC layers carry out the last classification task. Many pre-trained models (i.e., GoogLeNet, ResNet, and VGG-16) have been developed recently for CNN feature-map extraction, including EfficientNet, which enhanced network architecture and improved performance and speed.

The VGG16 architecture served as the foundation for our CNN model in this investigation. The VGG16 is a popular DCNN model that is renowned for its consistent architecture. Its deep layers and small (3x3) convolutional filters allow for the extraction of rich features. It is quite successful for transfer learning, particularly when labeled data is scarce, because of its pre-trained weights on ImageNet.This study directly employs similar networks and weights as the baseline since the pretrained model of VGG16 provides a simple network that is practical to use. Since the main goal of this work was to enhance classification performance, lightweight topologies like MobileNet and SqueezeNet were not selected, despite the fact that they offer advantages in computational efficiency. On our dataset, preliminary tests showed that these lightweight networks performed worse than VGG16, with a significant decline in accuracy and F1-score (see Table 3). For our application, VGG16 thus offered the optimum balance between prediction accuracy and model complexity. Thirteen convolutional layers make up the 5 blocks of 21 layers in VGG-16. CNNs extract features unique to the target dataset after gradually learning general features from the first layers.

A sample sewing image is created by concatenating the feature map outputs from the last convolution layer. Without additional training, we employ similar VGG16 networks and weights in this procedure. Only the feature map from the final layer was applied, based on the application of 3x3 small filters in the successive convolution layers of VGG16. It was found that feature extraction from convolution layers close to the input layer was more spatially structured for visual comparison. But it comprises noise from the fabric texture's low-level feature. Or else, the deeper layers will show additional semantic features that are extracted from sewing stitches. Thus, we calculated the feature maps from each last convolution layer in order to compute a number of outputs and gain a better understanding of the feature maps. The stitch line shape is shown in the feature maps that were taken from the layers close to the input layer. Also, the compression was applied to the remaining feature maps that were taken from the deeper layers. Due to the loss of spatial data, they mostly include semantic elements that are challenging for humans to identify. This outcome shows that the sewing stitch can be extracted using the VGG-16 feature-map model without the need for an extra training phase.

3.3. DQN-Based Ad Classification

Following, the DQN model is employed for the classification and detection of online personalized ads and user preferences. The objective of advertising strategy is to minimize the negative impact of advertisements while also optimizing the user experience. The problem of online advertising in recommendation system is difficult, since (i) the action of advertising agent (AA) is complicated and comprises three sub-tasks: whether to interpolate an advertisement into the present rec-list, if so, which ad is best; Because the three sub-tasks are intrinsically connected, meaning that when the AA decides to interpolate an advertisement, the locations and candidate advertisements are interactive to optimize the return in online advertising system; and (iii) the AA must minimize the negative effects of advertisements on user experience while also optimizing ad revenue. To overcome these problems, we introduce a DQN framework. The processing of state and action features is introduced first, and then an optimization algorithm is used to demonstrate the suggested DQN architecture [20].

The *State* s_t contains the contextual information (text/images), *Action (A)* contains rec-list of current requests which classify the ad into sensitive or non-sensitive, recommend, or skip content, and the *Reward (R)*, which is based on classification accuracy and user feedback (ads browsing history or user records clicks, skips, reports). rec-list of current requests, and an ads browsing history or user records. The sequence of recommendations (or advertisements) that the user has perused is known as the recommendation (or ad) browsing history.

To capture users' sequential preference for suggestions and advertisements separately, we employ two RNNs with Gated Recurrent Units (GRU). The last hidden state of the RNN is applied to reflect the user's preference for recommendations. p^{rec} (or p_t^{ad}), whereas the RNN input is the feature of the user's recently visited ads. The app version, OS (Android or iOS), and feed type (scrolling down/up the screen) of the user's current request are all included in the contextual feature. c_t. We next convert the concatenated features of the L suggested items that will be shown in the current request into a lower-dimensional dense vector, $rec_t\ tanh\ (W_{rec} concat(rec_1, \cdots, rec_L) + b_{rec})$, which represents the rec-list of the present request.

$$s_t = concat(p_t^{rec}, p_t^{ad}, c_t, rec) \quad (1)$$

The advertisements that were browsed at time t will be added to the search history for creating p_{t+1}^{rec} and p_{t+1}^{ad}. The recommendation system provides rec_{t+1}, whereas c_{t+1} is dependent on the user's activity at $t+1$ time. For the action $a_t = (a_t^{ad}, a_t^{loc}) \in \mathcal{A}$, a_t^{ad} represents the feature of ad, and $a_t^{loc} \in \mathbb{R}^{L+1}$ shows the location to interpolate the ad selected. A user browses this rec-list ad and provides their feedback after the AA executes an action a_t at the state s_t, viz., ad interpolating into a rec-list (or not). As a result, the immediate reward $r_t(s_t, a_t)$ is two-fold: (i) the user experience r_t^{ex}, and (ii) the classification of ad r_t^{ad}.

In a real-time platform, the primary threat of interpolating ads too frequently or improperly is that the user will leave the platform. After browsing the present rec-ad list, user experience can be measured whether by whether he/she will leave the platform, and we have:

$$r_t = \begin{cases} +2 & if\ classiciation\ correct\ AND\ positive\ user\ interaction \\ +1 & if\ classiciation\ correct\ AND\ positive\ user\ interaction \\ -1 & if\ misclassfied\ ,no\ negative\ feedback \\ -2 & if\ misclassfied\ with\ negative\ feedback \end{cases} \quad (2)$$

Let r_t be the reward at t time, determined by classification correctness and user engagement. In other words, if the user keeps looking through the following list, the AA will get a positive interaction; if not, they will get a negative user feedback. In the following, the reward function can be represented as:

$$r_t(s_t, a_t) = r_t^{ad} + \alpha \cdot r_t^{ex} \quad (3)$$

Here, the r_t^{ad} denotes the reward of ad, viz., a positive value if an ad interpolated, or lse 0. The hyperparameter α controls the significance of the second term, where the ad influence can be measured based on user experience. The optimal action-value $Q^*(s_t, a_t)$ with parameter θ is given as:

$$Q^*(s_t, a_t) = E_{s_{t+1}}[r_t + \gamma \max Q^*(s_{t+1}, a_{t+1}) | s_t, a_t] \quad (4)$$

The algorithm of DQN is given below.

Pseudocode of DQN model

Initialize Q action value with randomized weights
for session $= 1, M$ do
Initialize s_0 state from earlier session
for $t = 1, T$ do
Observe state $s_t = concat(p_t^{rec}, p_t^{ad}, c_t, rec)$
Execute action a_t
Calculate reward $r_t = r_t^{ad} + \alpha r_t^{ex}$ from offline log
Update state s_t to s_{t+1}
Store transition (s_t, a_t, r_t, s_{t+1}) to $\mathcal{D}$
Sample mini-batch of transitions (s, a, r, s') from $\mathcal{D}$
Set $y = \begin{cases} r & terminal\ s' \\ r + \gamma \max_{a'} Q(s', a'\ ;) & nonterminal\ s' \end{cases}$
Minimize $(y - Q(s, a; \theta))^2)$
End for
End for
Return: Trained Q-network $Q(s, a;\ \theta)$

3.4. Oputna Using Hyperparameter Tuning

Finally, the hyperparameter tuning using Optuna optimization algorithm is performed for improving the real-time classification performance. As an advanced hyperparameter optimization framework, Optuna choses the optimum integration of hyperparameters by enhancing a predetermined objective function. The optimization process, individual trials, and the objective function are the main elements of the framework [20]. Optuna uses sampling-based techniques and pruning algorithms to choose hyperparameters, which successfully reduces the model fitting time while enhancing performance, in contrast to conventional grid search methods. In this study, more accurate and effective landslide identification can be enabled by the incorporation of the RF algorithm with the Optuna framework. In the Optuna hyperparameter optimization process, geomatics, natural hazards, and risk are four primary processes. (1) Define the objective function, which is to specify the range of hyperparameters for the classification model, and maximize the F1 score; (2) Use the provided hyperparameters to train the classification model in each trial, make predictions based on the validation data, and determine the F1 score. (3) To maximize the classifier's performance, perform several trials to determine the hyperparameters that produce the highest F1 score. (4) produce the optimal hyperparameter together with the associated F1 score.

The Optuna optimization algorithm increases the fitness function (FF) to achieve superior classification outcomes. It recognizes a positive integer to symbolize the greater outcome of the candidate solutions. In this study, the decline of classification error rate can be observed as FF, as given below.

$$fitness(x_i) = ClassifierErrorRate(x_i)$$

$$= \frac{no.of\ misclassified\ samples}{Total\ no.of\ samples} * 100 \quad (2)$$

4. Results and Discussion

The experimental analysis of the HCOAUICNN-DQN approach was implemented using Python and TensorFlow. Training and evaluation were performed on a workstation equipped with an NVIDIA RTX 3080 GPU and 64GB of RAM. The model's performance can be evaluated through accuracy, precision, recall, and f1-score. The HCOAUICNN-DQN method was examined by using a Kaggle dataset [21] comprising 1175 samples under 2 class labels with various classes as shown in Table 1.

Table 1. Details of the Dataset

Category	No. of Samples
Sensitive Content	1367
Non-Sensitive Content	408
Total No. of Samples	1775

The confusion matrices attained by the HCOAUICNN-DQN technique with 80:20 and 70:30 of TRPH/TSPH are shown in Figure 2. The outcome refers to the effective detection and classification of all classes. Both TR and TS phases were used to assess the model's robustness in various data partitioning scenarios.

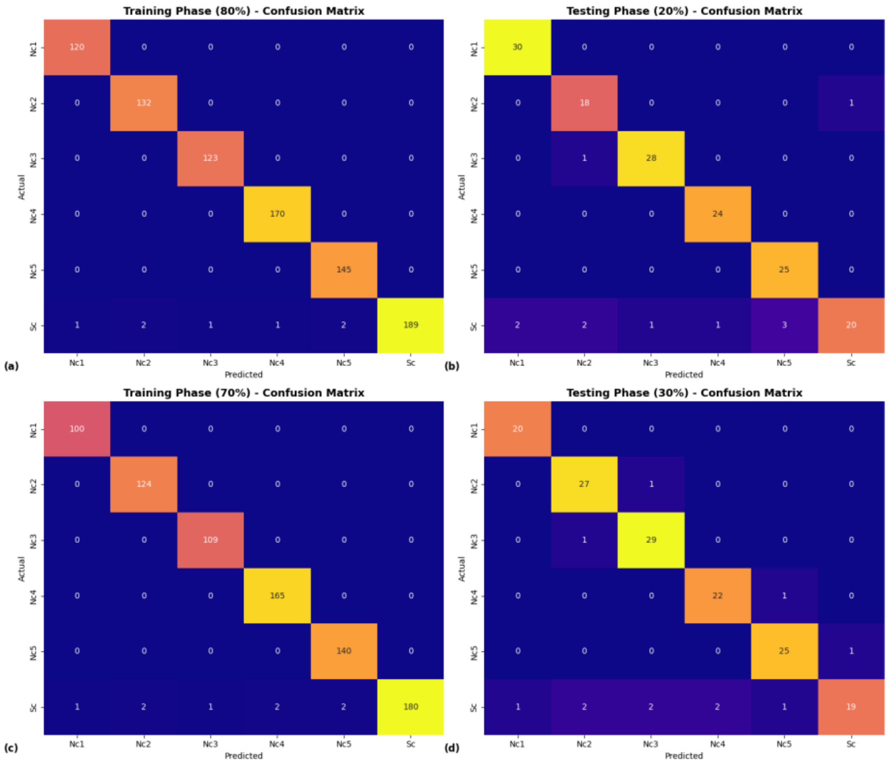


Fig 2. Confusion matrices of HCOAUICNN-DQN method under (a-b) 80:20 and 70:30 (c-d) TRPH and TSPH

The confusion matrices of the HCOAUICNN-DQN approach on the Ad classification process are illustrated in Figure. 2. On 80% of TRPH, the HCOAUICNN-DQN model classifies 690 instances into the SC class and 189 instances into the NSC class. Moreover, on 20% of TSPH, the HCOAUICNN-DQN approach classifies 125 instances into the SC class and 25 instances into the NSC class. As well, on 70% of TRPH, the HCOAUICNN-DQN system classifies 638 instances into the PD class and 180 instances into the NPD class. At last, on 30% of TSPH, the MPADLB-PDC method classifies 123 instances into the SC class and 19 instances into the NSC class.

The overall online ad classification results of the HCOAUICNN-DQN methodology are demonstrated in Table 2 and Figure 3. The experimental validation stated that the HCOAUICNN-DQN approach obtained a better classification process. For instance, on 80%of TRPH, the HCOAUICNN-DQN approach attains average $accu_y$ of 98.83%, $prec_n$ of 97.90%, $reca_l$ of 96.93%, F_{score} of 97.92%. Simultaneously, on 20% of TSPH, the HCOAUICNN-DQN system attains average $accu_y$ of 98.12%, $prec_n$ of 97.29%, $reca_l$ of 94.993%, F_{score} of 99.41%.

Table 2. Ad classification results of HCOAUICNN-DQN method under various metrics

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
Training Phase (80%)				
Sensitive Content	99.90	98.22	96.70	98.96
Non-Sensitive Content	99.76	97.59	97.16	96.87
Average	99.83	97.90	96.93	97.92
Testing Phase (20%)				
Sensitive Content	98.19	97.72	95.18	88.44
Non-Sensitive Content	98.12	96.86	94.68	86.36
Average	98.12	97.29	94.93	99.41
Training Phase (70%)				
Sensitive Content	97.81	96.07	98.41	96.52
Non-Sensitive Content	94.46	95.52	97.46	95.01
Average	96.14	95.80	97.94	95.77
Testing Phase (30%)				
Sensitive Content	96.48	95.05	96.68	95.73
Non-Sensitive Content	95.57	95.03	96.57	94.41

Average	96.03	95.04	96.63	95.07

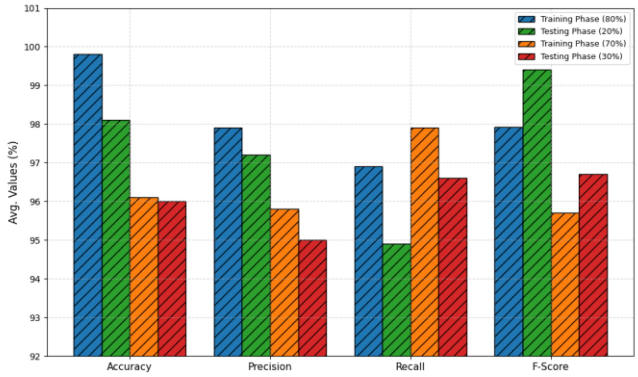


Fig 3. Average outcome of the HCOAUICNN-DQN system with distinct measures

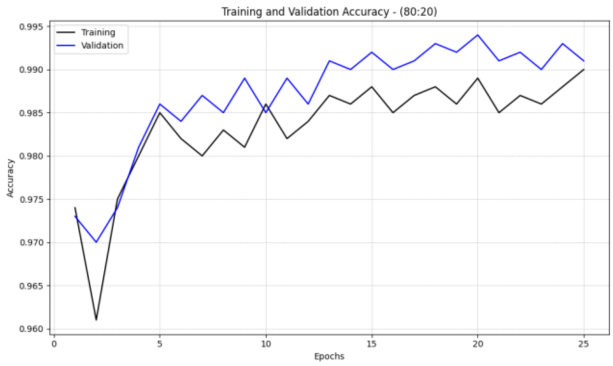


Fig 4. Accuracy curve of the HCOAUICNN-DQN system on 80%of TRPH/20% of TSPH

Figure 4 examines the accuracy of the HCOAUICNN-DQN model in the training and validation procedure on 80%of TRPH/20% ofTSPH. The outcome inferred that the HCOAUICNN-DQN methodology gains higher accuracy values over increased epochs. Moreover, the maximum VLAC over TRAC demonstrates that the HCOAUICNN-DQN algorithm learns capably on 80:20 of TRPH/TSPH. The curves for TRAC and VLAC provide valuable insights into their effectiveness at different epochs. Especially, it becomes a constant upgrading in the TRPH and TSPH with increased epochs, indicating the model's ability to learning and recognizing patterns at both TR and TS datasets The upward tendency in TS emphasizes the adaptability of model for the dataset of TR as well as its capabilities to make correct predictions on unobserved dataset, emphasizing abilities of strong generalization.

A wide-ranging overview of the TRLS and VLLS values for the HCOAUICNN-DQN method on 80%of TRPH/20%TSPH across several epochs is shown in Figure 5. The TRLS consistently decreases as the model refines its weights to minimize classifier errors on TR and TS datasets. The TRLS and VLLS curves clearly demonstrate the model's alignment with the TR dataset, underscoring its proficiency to capture patterns effectively in both datasets. There is the continuous improvement of parameters in the HCOAUICNN-DQN method, aimed at reducing inconsistencies between predictions and actual TR labels.

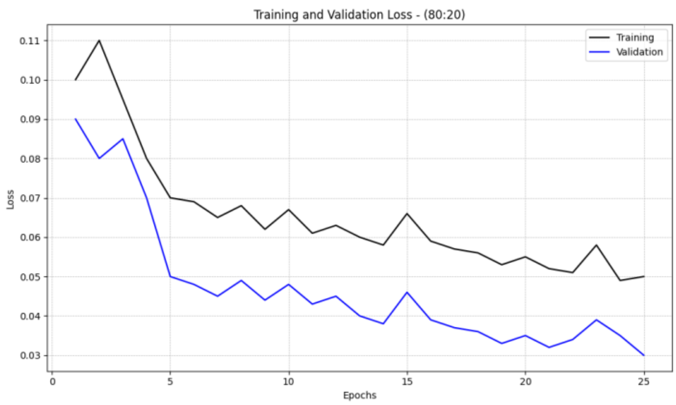


Fig 5. Loss curve of the HCOAUICNN-DQN system on 70% of TRPH/30% of TSPH

The curves for TRAC and VLAC established in Figure 6 for the HCOAUICNN-DQN approach at 70% of TRPH/30% of TSPH provide respective insights into its efficiency under various epochs. Particularly, there is a consistent improvement in both TR and TS datasets with improving epochs, specifying the model's adaptability in learning and identifying patterns in both datasets. The increasing tendency in TSPH emphasizes the model proficiency to the TR dataset along with its capabilities for creating correct predictions on unnoticed data, highlighting the abilities of robust generalization.

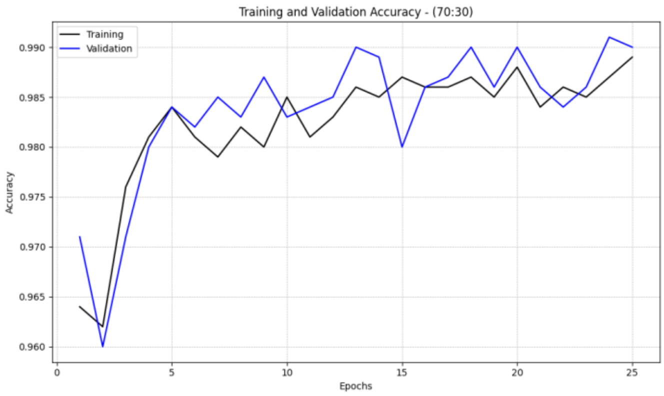


Fig 6. Accuracy curve of the HCOAUICNN-DQN system on 70:30 of TRPH/TSPH

A comprehensive review of the TR and TS loss values for the HCOAUICNN-DQN approach on 70% of TRPH/30% of TSPH through varying epochs is shown in Figure 7. The TR loss dependably decreases as the model refines its weights for reducing classifier error rates on these datasets. The loss curves characterize the model's alignment with the TR dataset, underlining its capability for capturing patterns effectively than other models. Notably, there is the consistent improvement of parameters in the HCOAUICNN-DQN approach, targeting at minimizing inconsistencies amongst predictions and actual TR labels.

Fig 7. Loss curve of the HCOAUICNN-DQN system on 70% of TRPH/30% of TSPH

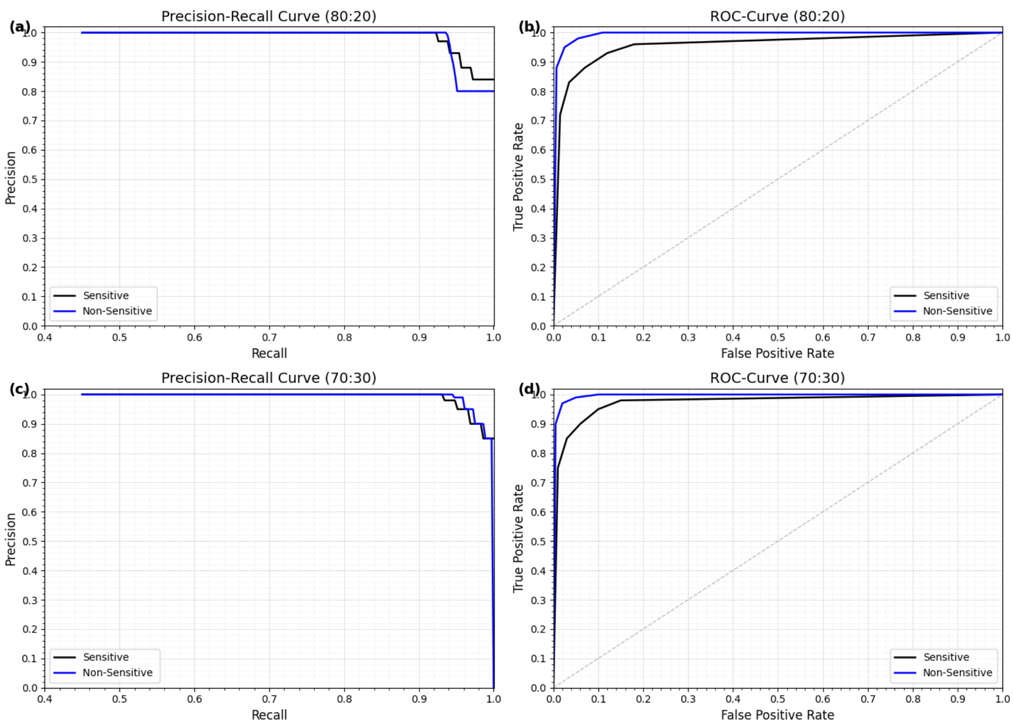


Fig 8. PR curve on 80:20 (a-c) and 70:30, and ROC curve on 80:20 and 70:30 (b-d)

Figure 8 illustrates the detection analysis of the HCOAUICNN-DQN method at 80:20 and 70:30. Figs. 7a-7c provides the PR efficiency of the HCOAUICNN-DQN technique. The experimental results achieved infer that the HCOAUICNN-DQN method attains improved values of PR. Additionally, it is noted that the HCOAUICNN-DQN method gets maximum PR values with two classes. Also, Figures. 7b-7d exhibits the ROC analysis of the HCOAUICNN-DQN approach. This figure implies that the EWC-DLFA algorithm acquires boosted ROC values. Also, the HCOAUICNN-DQN model offers maximum ROC values with two classes.

The Ad classification outcomes of the HCOAUICNN-DQN method undergo comparison with recent DL techniques in Table 3 and Figure 9. From the outcomes, it can be seen that the MPADLB-PDC algorithm exhibits outperforming outcomes over other approaches. With respect to $accu_y$The HCOAUICNN-DQN technique offers improved $accu_y$ of 99.83%, while the AlexNet, VGG-16, RNN, and EfficientNet-BiLSTM models attain decreased $accu_y$ of 98.43%, 97.90%, 96.15%, and 98.94% correspondingly. Simultaneously, based on $prec_n$, the HCOAUICNN-DQN method offers enhanced $prec_n$ of 97.90%, while the AlexNet, VGG-16, RNN, and EfficientNet-BiLSTM models realize minimal $prec_n$ of 95.59%, 96.35%, 89.23%, and 97.51% correspondingly. Concurrently, in terms of $reca_l$, the HCOAUICNN-DQN system offers superior $reca_l$ of 96.93%, while the AlexNet, VGG-16, RNN, and EfficientNet-BiLSTM systems achieve lower $reca_l$ of 96.43%, 95.55%, 88.03%, and 96.65% correspondingly. Finally, based on F_{score}, the HCOAUICNN-DQN approach provides higher F_{score} of 97.92%, while the AlexNet, VGG-16, RNN, and EfficientNet-BiLSTM methodologies accomplish decreased F_{score} of 94.41%, 96.98%, 95.06%, and 93.80% correspondingly.

Table 3. Comparative outcome of the HCOAUICNN-DQN with other DL approaches

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
AlexNet	98.43	95.59	96.43	94.41
VGG-16	97.90	96.35	95.55	96.98
RNN	96.15	89.23	88.03	95.06
EfficientNet-BiLSTM	98.94	97.51	96.65	93.80
HCOAUICNN-DQN	**99.83**	**97.90**	**96.93**	**97.92**

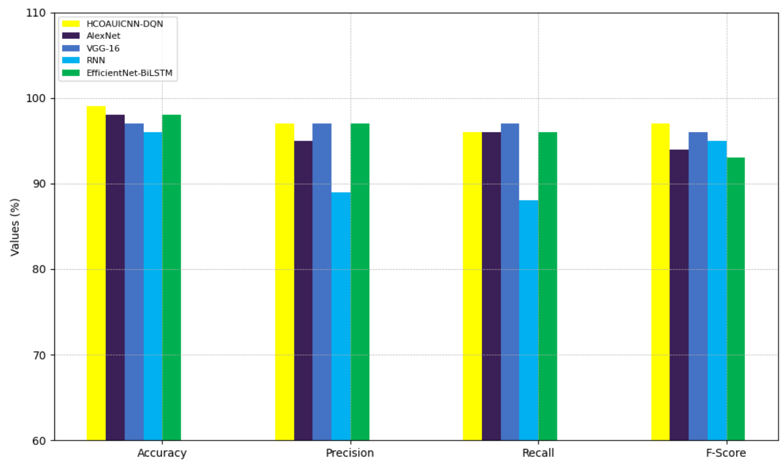


Fig 9. Comparative outcome of the HCOAUICNN-DQN method with other DL techniques

Consequently, the HCOAUICNN-DQN system exhibited maximum results on the online Ad classification method.

5. Conclusion

In this work, we present an efficient HCOAUICNN-DQN model. Initially, the HCOAUICNN-DQN model uses data preprocessing to preprocess the input dataset from the online advertisement website. Next, extract those input features through the pretrained CNN-feature maps. Following, DQN for the classification of online personalized ad user intent detection. Finally, the hyperparameter tuning using Optuna optimization algorithm is applied to improve the real-time classification performance. A set of experiments was conducted to analyze the model's efficiency using different datasets of advertising images. The performance of the HCOAUICNN-DQN model will be assessed through accuracy, f1-score, recall, and precision metrics. The HCOAUICNN-DQN method obtains superior outcomes when compared to other existing approaches under various metrics. In the future, the fairness-aware training methods should be investigated to lessen possible bias in ad classification. Decisions on sensitive content may become more transparent if explainable AI techniques like SHAP are used. For useful system improvement, real-time deployment and user input integration will also be examined.

References

[1] N. Nickel, “An investigation of how personal sensitive information impact the effectiveness of targeted online advertising,” Bachelor's thesis, University of Twente, 2024.

[2] M. Braun and E. M. Schwartz, “Where A/B testing goes wrong: How divergent delivery affects what online experiments cannot (and can) tell you about how customers respond to advertising,” *J. Mark.*, vol. 89, no. 2, pp. 71–95, 2025.

[3] L. Madio and M. Quinn, “Content moderation and advertising in social media platforms,” *J. Econ. Manag. Strategy*, vol. 34, no. 2, pp. 342–369, 2025.
[4] Z. Pooranian, M. Conti, H. Haddadi, and R. Tafazolli, “Online advertising security: Issues, taxonomy, and future directions,” *IEEE Commun. Surv. Tutor.*, vol. 23, no. 4, pp. 2494–2524, 2021.
[5] I. Wooton and Z. Cui, “The effect of online advertising on consumer buying interest in online selling applications with customer satisfaction as an intervening variable (study from member of United Kingdom medical doctor department),” *Medalion J.: Med. Res., Nurs., Health Midwife Particip.*, vol. 3, no. 3, pp. 82–100, 2022.
[6] S. Chandra, G. Ghule, S. M. Bilfaqih, A. Thiyagarajan, J. Sharmila, and S. Boopathi, “Adaptive strategies in online marketing using machine learning techniques,” in *Digital Transformation Initiatives for Agile Marketing*, IGI Global, pp. 67–100, 2025.
[7] D. Austin, A. Sanzgiri, K. Sankaran, R. Woodard, A. Lissack, and S. Seljan, “Classifying sensitive content in online advertisements with deep learning,” *Int. J. Data Sci. Anal.*, vol. 10, no. 3, pp. 265–276, 2020.
[8] Y. Wu, “Creation, consumption, and control of sensitive content,” *Mark. Sci.*, vol. 43, no. 4, pp. 885–902, 2024.
[9] B. Berlilana, T. Hariguna, and I. M. El Emary, “Enhancing digital marketing strategies with machine learning for analyzing key drivers of online advertising performance,” *J. Appl. Data Sci.*, vol. 6, no. 2, pp. 817–827, 2025.
[10] E. K. Linardi, H. F. Lin, and B. Yeo, “Effective digital advertising: The influence of customised ads, self-esteem and product attributes,” *J. Creative Commun.*, vol. 19, no. 2, pp. 197–216, 2024.
[11] P. Jain, K. Taneja, and H. Taneja, “Convolutional neural network based advertisement classification models for online English newspapers,” *Turk. J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 1687–1698, 2021.
[12] T. Beauvisage, J.-S. Beuscart, S. Coavoux, and K. Mellet, “How online advertising targets consumers: The uses of categories and algorithmic tools by audience planners,” *New Media Soc.*, vol. 26, no. 10, pp. 6098–6119, Oct. 2023, doi: 10.1177/14614448221146174.
[13] V. K. Polina and K. Malathi, “Classification of target customers for online advertising using Wide ResNet CNN and comparing its accuracy over ELM-CNN algorithm,” in *Recent Research in Management, Accounting and Economics (RRMAE)*, Routledge, pp. 413–417, 2025.
[14] T. Palczewski and A. Rao, “Session-aware graph neural network-based recommendations for custom advertisement segment generation,” in *Proc. 2023 15th Int. Conf. Mach. Learn. Comput.*, Feb. 2023, pp. 141–145.
[15] B. Joy and V. R. Deepthi, “A tensor based approach for click fraud detection on online advertising using BiLSTM and attention based CNN,” in *2023 Int. Conf. Self Sustain. Artif. Intell. Syst. (ICSSAS)*, IEEE, Oct. 2023, pp. 669–674.
[16] A. Erkan and T. Güngör, “Analysis of deep learning model combinations and tokenization approaches in sentiment classification,” *IEEE Access*, vol. 11, pp. 134951–134968, 2023.
[17] C. Zhou et al., “Atrank: An attention-based user behavior modeling framework for recommendation,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2023.
[18] L. D. F. Santos and M. V. da Silva, “The effect of stemming and lemmatization on Portuguese fake news text classification,” *arXiv preprint* arXiv:2310.11344, 2023.
[19] H. Chowdhury et al., “Broken stitch detection method for sewing operation using deep learning,” in *2024 Int. Conf. Innovations Sci., Eng. Technol. (ICISET)*, IEEE, pp. 1–6, Oct. 2024.
[20] X. Zhao et al., “DEAR: Deep reinforcement learning for online advertising impression in recommender systems,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, pp. 750–758, May 2021.
[21] N. Aniruddhan, “Online advertising digital marketing data,” *Kaggle*, 2023. [Online]. Available: https://www.kaggle.com/datasets/naniruddhan/online-advertising-digital-marketing-data.