IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 September 2022, accepted 26 September 2022, date of publication 3 October 2022, date of current version 3 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211528

RESEARCH ARTICLE

An Ensemble Architecture Based on Deep Learning Model for Click Fraud Detection in Pay-Per-Click Advertisement Campaign

AMREEN BATOOL[1] AND YUNG-CHEOL BYUN[2]

[1]Department of Electronic Engineering, Institute of Information Science and Technology, Jeju National University, Jeju 63243, South Korea
[2]Department of Computer Engineering, Major of Electronic Engineering, Institute of Information Science and Technology, Jeju National University, Jeju 63243, South Korea

Corresponding author: Yung-Cheol Byun (ycb@jejunu.ac.kr)

This work was supported in part by the Ministry of Small and Medium-Sized Enterprises (SMEs) and Startups (MSS), South Korea, under the "Regional Specialized Industry Development Plus Program (Research and Development)" supervised by the Korea Institute for Advancement of Technology (KIAT), under Grant S3246057; and in part by KIAT Grant by the Korea Government (MOTIE) (The establishment project of industry-university fusion district), under Grant P0016977.

ABSTRACT With the rapid development of online advertising, click fraud is a serious issue for the internet market. Click fraud is a dishonest attempt to improve a website's profit or deplete an advertiser's budget by clicking on pay-per-click advertisements. For an extended period, this illegal act has a threat to the industrial sectors. As a result, these businesses hesitate to advertise their items on mobile apps and websites, as numerous groups attempt to take advantage of themes. To safely advertise their services and products online, a robust mechanism is needed for efficient click fraud detection. To tackle this issue, an ensemble architecture of machine learning and deep learning is proposed to detect click fraud in online advertisement campaigns. The proposed ensemble architecture consists of a Convolutional Neural Network (CNN), and a Bidirectional Long Short-Term Memory network (BiLSTM) is used to extract hidden features, while the Random Forest (RF) is used for classification. The main objective of the proposed research study is to develop a hybrid DL model for automatic feature extraction from clicks data and then process through an RF classifier into two classes, such as fraudulent and non-fraudulent clicks. Furthermore, a preprocessing module is developed to preprocess data by dealing with categorical attributes and imbalanced data to enhance the reliability and consistency of the clicks data. In addition, different evaluation criteria are used to evaluate and compare the performance of the proposed CNN-BiLSTM-RF with the ensemble and standalone models. The experimental results indicate that our ensemble architecture achieved the accuracy of $99.19 \pm 0.08\%$, precision $99.89 \pm 0.03\%$, sensitivity $98.50 \pm 0.11\%$, F1-score $99.19 \pm 0.08\%$ and specificity $99.89 \pm 0.03\%$. Furthermore, our proposed architecture produced superior results compared to other developed ensemble and conventional models. Moreover, our proposed ensemble architecture can be used as a safeguard against click fraud for pay-per-click advertising to facilitate industries for the safe and reliable promotion of their products.

INDEX TERMS Online advertising, pay-per-click, click fraud, machine learning, deep learning, ensemble learning.

I. INTRODUCTION

Online advertising, often known as online marketing or Internet advertising, is one of the most profitable and rapidly growing industries [1]. It utilizes the Web to disseminate products, services, and viewpoints for marketing or brand promotion [2]. Online advertising is the crucial source of revenue for internet giants like Yahoo, Google, and Facebook [3]. These giants are ad networks that serve as intermediaries between publishers and advertisers.

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masucci.

Advertisements are delivered to these ad networks, which agree on a pre-decided price for each user click. The ad network compensates the content publisher based on the number of visitors it directs to the ads [4]. Unfortunately, a threat known as Click Fraud is associated with this payment method. There is roughly one fraudulent click among five clicks. The practice of click fraud is becoming more common, and a significant part of internet traffic is fake. In addition, advertisers typically suffer from economic losses due to click fraud activities [5].

Click fraud can be defined as deliberately clicking on a pay-per-click advertisement to redirect or negatively use the advertiser's ad budget. [6], [7]. Numerous groups or parties engage in click fraud. The most frequent offenders are as follows: Competitors engage for the largest share of click fraud in their competitor's adverts. They generate clicks and acquire a competitive edge by squandering their opponent's pay-per-click budget. Web administrators conduct click fraud and make unjustifiable revenue from displaying advertisements on their websites. Rather than spending time growing and improving their website, they are enticed to click on these advertisements to gain profits. A fraud ring is an organized group of fraudsters who target ad networks to obtain more money quickly [8].

Click fraud may be performed in a variety of ways. A brute force attack is an approach to click fraud using a single computing device. This attack might be as simple as repeatedly clicking on an advertisement. Publishers employ crowd-sourcing to boost ad clicks. They intentionally or unintentionally use website users to click on their ads. Reward traffic compensates the user for clicking on the ad, a more sophisticated form of click fraud that can create many clicks. Click Farm is a click fraud technique that persuades people to click on advertisements for a whole day in return for money. Hit Inflation is another type of click fraud in which real users are redirected to a website by visiting the ad and then the page they want to see. Botnets are malware that spreads through a network of infected computers. Malware takes control of several computers. They tell the hacked computers to browse various websites and click on their advertising without the owner's knowledge [9].

Recently, machine learning (ML) and deep learning (DL) paradigms are used widely for online fraud detection, such as user behaviors and items fraud [10], tax fraud [11], financial and transaction [12], [13], credit card fraud detection [14], [15], [16], to name a few. The conventional ML models, such as Random Forest (RF), Support Vector Machines (SVM), Naive Bayesian (NB), etc., rely on the manual representation of features space, which require human intervention to construct features space before the learning process. Furthermore, these conventional models are also not adaptable to cope with high-dimensional data [17]. To cope with this issue, DL models, such as Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) are robust and provide automatic feature construction from large data samples [18], [19], [20]. These models are also leveraged to cope with high dimensional and non-linear data to produce reliable and superior performance compared to the conventional learning models [21], [22]. Motivated by these studies, therefore, in this study, an ensemble architecture is proposed based on a two-fold hybrid approach to classify normal and fraudulent clicks. First, DL models, such as CNN and BiLSTM, are combined to develop a robust architecture to construct feature space automatically. Second, RF is employed as a supervised learning model to take features constructed using a hybrid DL model to classify clicks into two classes, such as normal and fraudulent.

The following are the major contributions of this study.

- Develop an ensemble CNN-BiLSTM-RF architecture based on ML and DL models to detect click fraud in online advertisement campaigns with high accuracy.
- Develop a pre-processing module to investigate temporal characteristics of the dataset for click fraud detection.
- Comparative analysis of the proposed ensemble CNN-BiLSTM-RF architecture with the conventional learning and DL models to highlight the significance of the proposed research study.
- Various experiments are performed as a proof of concept to highlight the significance of the proposed ensemble architecture to facilitate the advertisement industry for reliable product promotions.
- A complete experimental study is provided to assess the performance of the proposed model, including Area Under Curve (AUC), precision, specificity, sensitivity, F1-score, confusion matrix and accuracy.

The remaining flow of our proposed architecture is structured as In section II discussed the latest and most related work. Section III illustrates the proposed architecture methodology and materials. Data description, pre-processing and analysis are given in section IV. Section V contains the details experimental setup and results analysis. Section VI compared the proposed approach with other latest methods and section VII conclude the article.

II. RELATED WORK

This section discusses the existing approaches for click fraud detection in Pay-per-Click advertising campaigns. Many studies exist in the literature focused on click fraud detection employing machine learning and deep learning approaches. However, this section discusses a few recent and related studies. In [23], proposed Clicktok, based on unifying the technical response and exploiting the temporal aspects of click traffic, provides a protection approach that separates organic and non-organic click fraud attacks.To identify the online click request author used AdSherlock and detected click fraud after that [24]. In [25], a robust integrated local kernel embedding model was proposed to handle data sparsity and imbalance problems through a robust similarity function, which obtains the data embedding. In [26], presented an efficient and deployable solution for detecting click fraud at the client side in mobile apps. Finally, in [27], the Fight

Click-Fraud (FCFraud) method was proposed to detect click fraud from the user side, which can be incorporated into smartphone and computer operating systems. The proposed method accurately classifies ad requests from all user actions 99.6% accurately detects click bots 100% successfully on mobiles and computer devices.

Different supervised learning models have been developed to detect click fraud in an online advertisement environment. In [28], proposed an ad-fraud-detection approach that utilizes robust features against attacker evasion. In [29], the authors developed new features based on statistics seen in an ad network, estimated from a considerable number of legitimate user ad requests, including the popularity of publisher websites and client environment tendencies [30]. These features are fed to the RF for detecting fraudulent ad requests. In [31], assessed the user's click journey across their portfolio and flagged IP addresses that generate many clicks but never install apps. They used LightGBM as a methodology and achieved 98% accuracy. In [32], the authors proposed a model based on XGBoost for distinguishing between legal and illegal users. In [33], analyzed click patterns across a dataset to determine the user's click journey across their portfolio and fagged IP addresses that generate a high number of clicks but not complete installation of the app. They used SVM, KNNs, RF, and Gradient Tree Boosting (GTB) for classification. In [6], the RF algorithm was used to classify features to predict fraudulent click behaviour and achieved prediction accuracy higher than 91% in the positive and negative samples. However, all these models require a manual process to construct features from the given data samples. Furthermore, these conventional learning models are not versatile enough to cope with high dimensional and non-linearity data problems, which degrade the model performance and cause poor generalization.

Furthermore, click fraud not only bothers budget advertisers but also demonstrates how bots are being used to tamper with your data. In [34], useful information, as a result, is critical to be aware of and evolve. It is to devise solutions to avoid and prevent them. In [35], the authors proposed a click fraud detection model, abbreviated CFC, for classifying fraudulent clicks by incorporating some features and testing with KNN, ANN, and SVM. The experiment results show that the proposed CFC model achieved more than 93%. In [36], Using ADASYN with GTB to over-sample the data enhanced the classification accuracy with an average precision score of 64.32% because the accuracy measure is not appropriate for the imbalance distribution of class samples. Therefore, the author used the F1 score and AUC as evaluation measures to assess the performance of GTB. In [37], an algorithm-based detection technique for classifying target advertising click frauds demonstrates how machine learning approaches can be integrated to maintain the viability of online advertising. In [38], CFC (Click Fraud Crowd-sourcing) Approach defend the dishonorable Clicks. In [39], the authors intended to detect clicks fraud using various ML and DL classifiers, such as RF, Support Vector Machines (SVM), k-Nearest Neighbor (KNN), as well as DL methods such as auto-encoders, Convolutional Neural Networks (CNN), Restricted Boltzmann Machine (RBM), etc. The limitation of this solution is that it only detects fraud in a supervised learning context.

Moreover ensemble strategies were also developed to detect click fraud to facilitate the advertising industry for reliable product promotions. In [40], the authors suggested an ensemble approach based on RF to classify ads impression into two classes, such as fraudulent or non-fraudulent. The authors achieved a precision of 96.29% using data acquired from a European commercial ad server. In [5], an ensemble approach was developed by integrating Cascaded Forest and XGBoost to detect click fraud using multiple datasets to evaluate the performance of the existing approach. The authors achieved the maximum precision of 94.0%, recall of 94.0%, F1-score 94.0%, and accuracy of 94.53%. Another Gradient Tree Boosting (GTB) based ensemble model was proposed to classify fraudulent behaviours of publishers from raw user click data [41]. The authors reported that the GTB model achieved a precision of 60.5%, which still needs improvement to facilitate industry for reliable online advertisement. Similarly, in [42], a two-fold strategy was developed to segregate non-human clicks from online advertisement camping data. Another ensemble model was presented in [43] to employ XGBoost for detecting click ad fraud using online advertising clicks data. The authors achieved an accuracy of 96% to facilitate advertisers to block fake ads for reliable product promotions.

In [44], a hybrid deep learning method comprised of a Neural Network (NN), Semi-supervised Generative Adversarial Network (GAN) and an Auto-Encoder (AE) was developed for click fraud detection. Furthermore, a multi-time scale forecasting technique was presented to deal with the imbalanced dataset. In [45], proposed a deep learning approach called the cost-sensitive CNN model to identify fraudulent clicks using mobile advertisement data based on the feature matrix of a click to capture the pattern of click fraud. As a result, they obtained a classification accuracy and recall rate of over 93%. In [46], they proposed a unique weighted hybrid model to detect click fraud and identify fraudulent mobile advertising apps by integrating heterogeneous graph, and DL approaches. The proposed approach is based on the mobile ad system's relationships among users, publishers and advertisements. Furthermore, Table 1 illustrates a critical analysis of the existing models for click fraud detection.

To the best of our knowledge, all aforementioned studies attempted to use either ML or DL models to detect click fraud in advertisement data. Furthermore, most of the studies employed manual approaches for feature extraction to detect click fraud detection. In addition, all these studies did not achieve accurate performance for click fraud detection to facilitate advertisement industries. To sum up, no study used an ensemble approach to automatically extract features using DL methods from given clicks data to classify into two real and fraudulent clicks using a supervised learning algorithm. Furthermore, existing clicks fraud detection models are failed

to achieve higher detection rate to facilitate advertisement industry for secure products promotions. Therefore, in this study, an ensemble CNN-BiLSTM-RF is developed to extract features automatically using a robust hybrid DL model and used RF classifier to classify real and fraudulent clicks. The proposed ensemble architecture aims to extract the most promising features automatically to build a robust classifier for enhancing the performance of the clicks fraud detection and also facilitating advertisement industries for reliable product promotions.

III. PROPOSED METHODOLOGY

This section presents a detailed methodology for click fraud detection architecture. It includes a general flow model of CNN architecture and detailed step-by-step architecture of the proposed method. In addition, it presents a comprehensive review of deep learning and conventional machine learning models.

A. OVERVIEW OF PROPOSED MODEL

This subsection shows an overview model of the proposed click fraud detection. The proposed model consists of the following steps as shown in the Figure 1. The first step illustrates the pay-per-click data, which includes real and fake click data. Next, raw clicks data is passed to the data cleaning module to cope with data imbalance problems, missing attribute values, selection of machine-readable features, etc. In the next step, prepared data is given as an input to the temporal features extraction module, which is responsible for extracting hidden temporal patterns from the given data. In addition, extracted features are visually analyzed to understand the hidden temporal patterns. Next, our prepared data is divided into training (learning) and testing (unseen) sample sets. Furthermore, our proposed CNN-BiLSTM+RF model is trained using the learning samples. Once our proposed model is trained, testing samples are passed to the learned model for evaluating the performance and performing a comparison with existing baseline models. To evaluate the performance of proposed and baseline models, different evaluation analysis matrices are used, such as accuracy, precision, recall, f1 score, and area under the curve.

B. STEP BY STEP PROCESS OF PROPOSED METHODOLOGY

In this subsection, a step-by-step process of the proposed Pay Per Click (PPC) methodology is discussed in Figure 3. The step-by-step process of the model illustrates several processes, including fake and real clicks data, processing the unprocessed data, extracting features, creating ML and DL models, and performance evaluation. First, we used the Synthetic Minority Oversampling Technique (SMOTE) for an unbalanced dataset to balance the dataset. Then, we divided the dataset into two classes real class and fake class. Finally, different performance matrices are considered to evaluate and contrast the effectiveness and performance of the proposed with the development approaches.

1) PRE-PROCESSING DATA

In pre-processing TakingData dataset has 200 million clicks per link including 8 feature. Data must be pre-processed before detection for the algorithm to recognize it.

- **Remove Null Value**: During data pre-processing, Machine Learning algorithms do not accept missing values; managing missing data is required during dataset preparation. We remove the missing values from the dataset using the distinct floating-point NaN value and the Python None object. i.e., removing rows that contain missing values. : Conversion of our data into numeric because in DL/ML, input and output variables are numeric, so we encode our categorical data into numeric data to fit and evaluate the model.
- **Transformation of Data**: Conversion of our data into numeric because in DL/ML, input and output variables are numeric, so we encode our categorical data into numeric data to fit and evaluate the model.
- **Data Re-sampling**: Data re-sampling is the prepossessing methodology to improve the accuracy of data. In our data, we used SMOTE oversampling technique to balance the distribution of the sampler by generating artificial samples to increase the samples of the minority class. The advantage of SMOTE is that it produces artificial data points rather than duplicates that are only slightly different from the actual data points.
- **Data Normalization**: It is a technique used in machine learning and deep learning to reduce the sensitivity of the training model to the number of features. It involves transforming real numerical value characteristics to a 0–1 range. It makes it possible for the model to converge to more precise weights. we used a common scale to change the value of numeric columns in our dataset because the features in the data have different ranges.

2) GENERAL CNN ARCHITECTURE

CNN is the improved version of multi-layer perception, proposed by [56]. CNN networks come in a variety of forms, including 1-dimensional (1-D) CNN, 2-dimensional (2-D) CNN, and 3-dimensional (3-D) CNN. We employed 1-D CNN in this study. The typical structure of 1-D CNN is visualized in Fig. 2.

CNN has three layers: a convolutional layer, a pooling layer, and a dense or fully connected layer. The CNN extracts implicit features from the input data by performing convolution and pooling operations [57]. The features gathered are then combined and routed through a dense or fully connected layer. An activation function is used to increase the non-linearity of neuron yield.

The convolutional layer is a critical component of the CNN architecture. By convolving the input data, a convolutional layer consists of several convolutional kernels that extract hidden features and build feature maps. Transmit the feature maps into a non-linear activation function to generate the convolutional layer output. Equation 1 is used to represent

TABLE 1. Critical analysis of existing clicks fraud detection models.

Model	Year	Main Objective	Techniques	Dataset	Findings	Limitations
RTMBM [47]	2018	Identified real mobile click bots through behavioral analysis	RF	Open Dataset	91.25%	Cannot detect placement click fraud using behavioral patterns.
Click Spam [48]	2018	Identified fraudulent clicks in a practical manner	KNN	Search engine log data (parsijoo.ir)	96%	Critical to identify user clicks
ML-FCD [49]	2019	Detected the click fraud in online environment	Light GBM	Public Dataset	98%	Unable to detect low-frequency nature of attacks.
Automated Bot [50]	2021	Identified click fraud now focuses on looking at server requests.	AdSherlock	Mobile Ads Data	61.3%	AdSherlock is produced low detection accuracy, which needs to be enhanced by combining the static analysis method.
ML-ETC [51]	2021	Comparative analysis to detect payment click fraud using ML classifiers.	Extra Trees Classifier	Talking and Mobile ads datasets	90%	The reported accuracy is still low, therefore, it is required to develop a robust model to enhance the performance.
HTM-CLA [52]	2021	Detected online credit card fraud in e-commerce systems	Simulated Annealing ANN	German and Australian CCF Datasets	85.71%	Unable to identify categorical data. Features reduction is also needed to minimize the computational cost.
CPC [53]	2021	Detected click fraud in online advertising systems.	GBDT	Real-World Ad Network Dataset	93%	Performance analysis is totally ignored to overcome click frauds in online ads.
LSH [54]	2022	Identified malicious clicks from an online advertisement	CNN	Talking Dataset	95%	Generalization issue for unseen clicks data.
DCNNTr [55]	2022	Identified user click on non image feature	DCNN	FDMA2012 user-clic	79.8%	The current work presented only few thousand sampled user clicks which makes learning too slow.

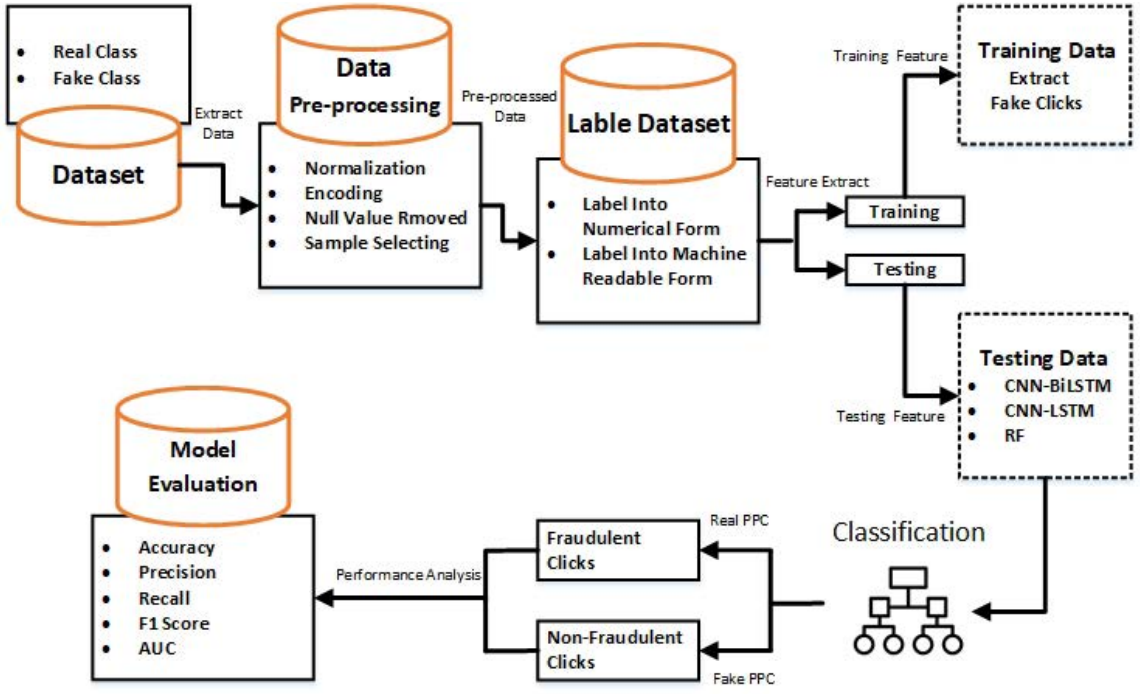


FIGURE 1. Overview model of the proposed methodology.

the convolutional layer mathematically, [58], that is.

$$c_j = f(w_j * x_j + b_j) \tag{1}$$

where x_j indicates the convolution layer input, c_j represents the j^{th} output feature map, w_j indicates a weight matrix, $*$ illustrates the dot product, b_j denotes the bias vector, and f indicates the activation function. As an activation function for CNN, rectified linear unit (ReLU) function is frequently used. Mathematically ReLU function can be defined as follows in Equation 2, [59]:

$$c_j = f(h_j) = max(0, h_j) \tag{2}$$

where the h_j represent a feature map element produced through convolutional methods. Pooling layers are also known as down-sampling. The pooling operation's primary function is to reduce the feature maps dimensionality and to avoid overfitting. The Max pooling layer is a popular pooling method. Mathematically can be calculated using Equations 3 and 4 to get the extreme value of an allocated area in feature maps [60].

$$\gamma(c_j, c_{j-1}) = f(h_j) = max(c_j - c_{j-1}) \tag{3}$$

$$P_j = \gamma(c_j, c_{j-1}) + \beta_j \tag{4}$$

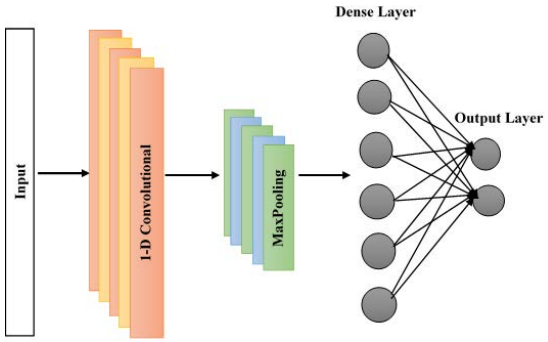


FIGURE 2. General flow model of CNN architecture.

where γ represents the maximum pooling sub-sampling function, β_j denotes the bias and P_j indicates the output of the max-pooling layer.

Finally, the convolutional and pooling feature maps are passed into the fully connected layer, which produces the ultimate output vector, which formulated by [60] as shown in Equation 5:

$$y_j = f(t_j P_j + \delta_j) \tag{5}$$

where y_j indicates the ultimate output vector, the bias represents with δ_j, and t_j denotes the weight matrix.

3) GENERAL BiLSTM ARCHITECTURE

BiLSTM is a bidirectional variant of LSTM used to learn in both directions, such as forward and backward, to process long data sequences [61]. The conventional LSTM often forgets future information, which causes a loss of information for long-term time-series data dependencies. The conventional LSTMs can also only use the prior context. Therefore, BiLSTM effectively employs two separate hidden layers to learn the long-term data sequences in forwarding and backward directions. It is preferable to capture two-direction contextual dependencies to gain access to long-range information. Simply, it consists of two LSTMs, where the first LSTM is employed to feed the learning process in the forward direction, whereas the second LSTM is used to learn from the given inputs in reverse (backward) direction.

Fig. 4 shows the basic architecture of the BiLSTM. $\overrightarrow{h}$ and $\overleftarrow{h}$ are used to individually indicate the output of the forward and reverse hidden layers. Both LSTM units use the ordered input data sequences in the training process. The recursive process is carried out to estimate the output of the forward $\overrightarrow{h_t}$ and reverse $\overleftarrow{h_t}$ LSTM layers. The output of both LSTM layers is merged using the mode attribute, whereas mode comprises the following possible merge strategies: average, sum, multiplied and concat. Our proposed architecture specifies an average mode to merge outcomes obtained from forward and reverse LSTM units. Finally, a flattened layer f_{layer} is employed to get the merged output and convert it into a one-dimensional vector v to obtain the desired outcome by passing vector v to the softmax function.

The layer of BiLSTM generates bi-directional sequences as an output two-dimensional vector, Y, where output sequences of both LSTM units are concatenated using merge mode strategy as shown in Equation 6, [18].

$$y_t = \alpha(\overrightarrow{h_t}, \overleftarrow{h_t}) \tag{6}$$

where the α indicates the merge mode strategy used for the both $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ output sequences. The α indicates an average mode strategy to concatenate the output sequences of both forward and reverse LSTM units. The merge mode strategy can be multiplication function, a summation function, an average function or a concatenating function. Finally, outcome of the both LSTM units is represented as a one-dimensional vector,$Y = [y_1, y_2, \ldots, y_t]$, where the last element, y_t, indicates the best-predicted value for the next time iteration.

4) WORKFLOW MODEL OF RANDOM FOREST

This subsection presents a general workflow of the conventional ML model, such as Random Forest (RF). The RF algorithm is a well-known supervised machine learning algorithm. RF is based on the ensemble learning concept, which is helpful for both regression and classification problems. It combines multiple classifiers to tackle a complex problem and improve model performance. As the name implies, RF is a classifier that uses several decision trees on different subsets of a dataset and gets the average to enhance classification accuracy. Rather than relying on a single decision tree, the random forest aggregates predictions from all trees and forecasts the ultimate output based on the majority vote of predictions. Higher the number of trees, the better the accuracy and the lesser the risk of over-fitting. This study using the classification to differentiate between real and fraudulent clicks. Thus, we utilize the primary binary RF classifier. Fig. 5 illustrates the architecture of a RF classifier.

C. PROPOSED CNN-BiLSTM-RF ARCHITECTURE

A broad overview of the developed approach is illustrated in Fig. 6. The proposed technique has three major components. The 11 × 1 Click fraud data is loaded into deep learning networks on the input layer. It comprises a 1-D CNN layer followed by a Maxpooling layer, allows for the sample-based discretization of parameters to recognize the relevant features resulting in reduced training time and prevention from over-fitting. After the Maxpooling layer comes the Batch Normalization layer, which enables the normalization of parameters between intermediate layers and prevents slower training times. The 1-D CNN layer contains 64 filters, kernel size two and Relu is used as an activation function. The Maxpooling layer is with pooling length 2. This feature map is fed into the BiLSTM layer. BiLSTM contains 128 memory blocks that learn the time domain features. The BiLSTM layer follows a Maxpooling layer with pooling length 2 and the Batch Normalization layer. Next, a Flatten to reshape the input for upcoming Dense layers. There are two dense layers

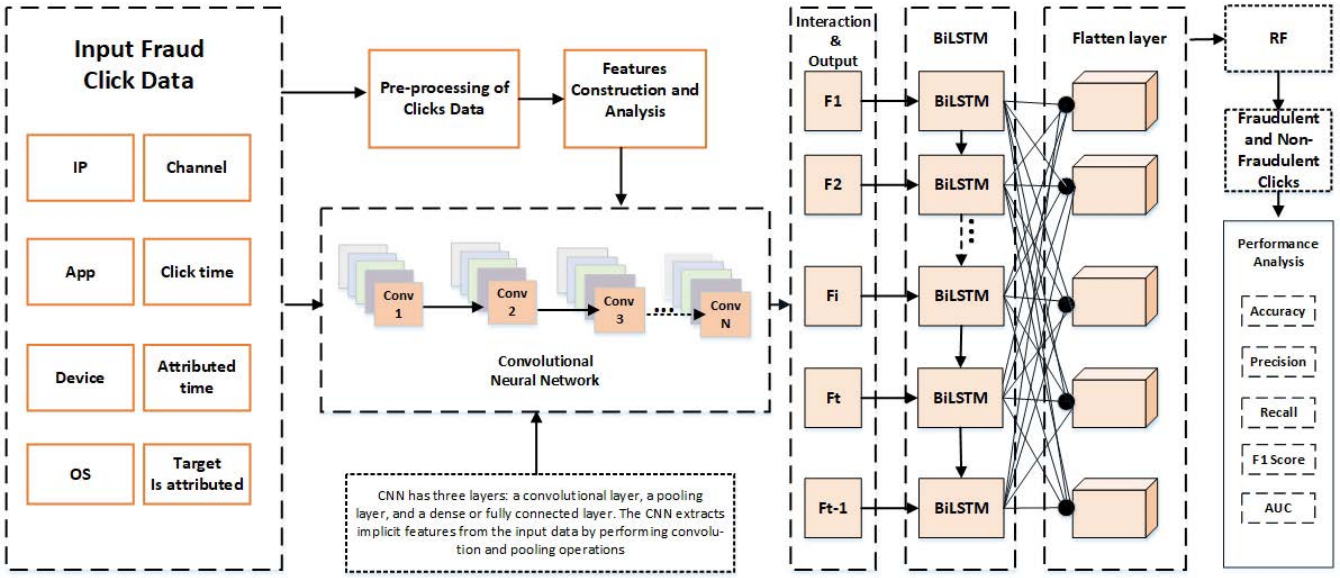


FIGURE 3. Overview model of the proposed methodology.

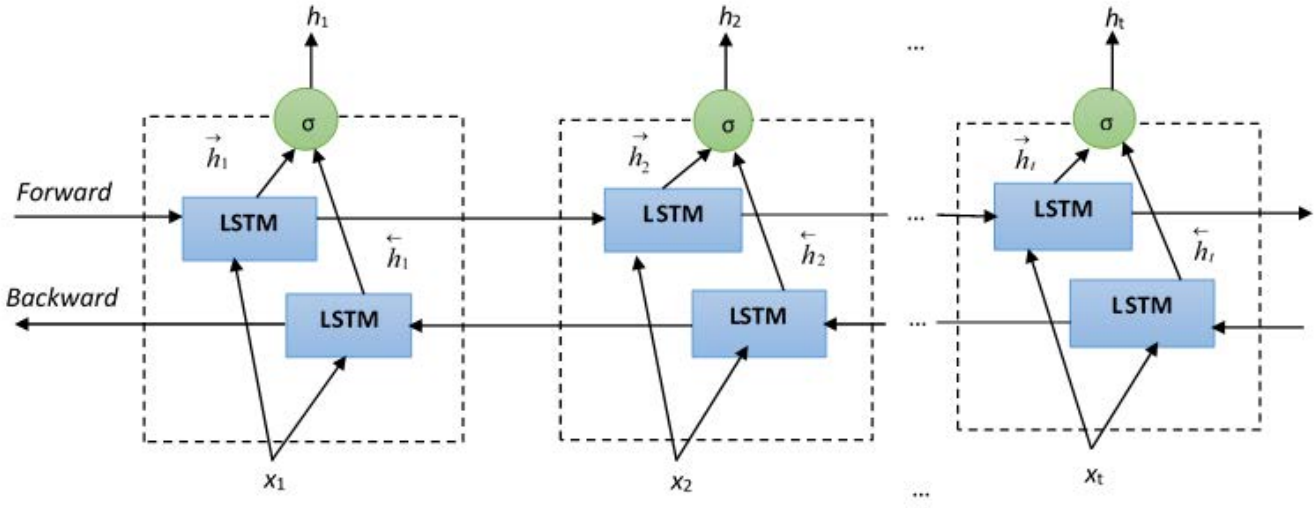


FIGURE 4. General flow model of BiLSTM architecture.

added with filters 128 and 64. Both dense layers have used Relu as an activation function. The dropout layer with a value of 0.5 is used between both dense layers. The Dropout Layer is put in place to account for Over Fitting even though the model uses Max Pooling in between every layer. Generally, this is because CNN and BiLSTM used in combination have a higher probability of over-fitting and perform poorly on the testing set. Finally, the features are fed into RF for the real and fraudulent clicks classification.

Furthermore, Algorithm 1 is presented to provide a step-by-step process of the proposed CNN+BiLSTM-RF. The proposed algorithm shows several steps to present a detailed flow of the model. It takes raw clicks data as an input and predict click type as a real or fake as an output. First of all, data is pre-processed in order to perform transmission of categorical attributes into number format, computational of initial input features using timestamp feature. Next, newly constructed features are added to the existing features set. Once features are combined, correlation index is calculated to reduce features set by eliminating low correlated features. Then, SMOTE is applied to balance the distribution of the data samples as per class label. Next, min-max normalization is used to scaled down features values in uniform range to consider all of the features equally in the learning phase of the learners. Once data is cleaned, in the next step, data is divided into $K(K = 10)$ subsets as $[S_1, S_2, S_3, \ldots, S_K]$.

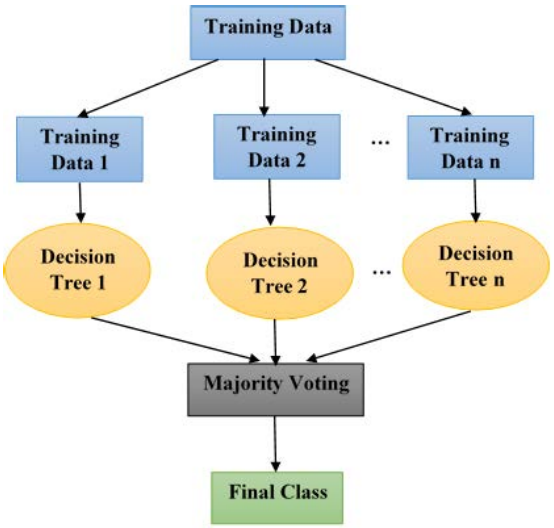


FIGURE 5. Basic workflow of the RF model.

Furthermore, an hybrid model is trained using $f(X, y)$ Where $X, y \in [S_1, \ldots S_K]$. During in each training epoch, training and validation loss and accuracy is computed and weights are updated for the next epoch using Adam optimizer to minimize training and validation error. In addition, for each K set, training and validation accuracy is reported. Once, hybrid CNN-BiLSTM model is trained, input samples are passed to the trained hybrid model to extract hidden features, which further used to train RF model using the $f(E_F, y_{train})$. Once RF is trained using extracted features set, unseen samples are also passed through the hybrid DL model to extract features for unseen samples using $f(\bar{E_F}, y_{test})$ to obtain y_{pred}. Once prediction results are obtained, different evaluation measures are employed to evaluate the performance in terms of Accuracy, Precision, Recall, F1 Score and AUC.

IV. DATASET PRESENTATION AND ANALYSIS

This section presents data preparation and analysis to clean the raw data to highlight the hidden patterns of clicks. It incorporates into data description with relevant data source, pre-processing of data in order to get reliable data,and exploratory analysis of pre-processed data.

A. DATASET DESCRIPTION AND PRE-PROCESSING

In this research study, we used the TalkingData dataset acquired from Kaggle [62], explained in detail as follows. The TalkingData is an ad-tracking fraud dataset containing 200 million clicks with eight features over four days. The main features of the dataset are as follows: click's IP address, application identifier for marketing purposes, device type, installed operating system for the device, publisher channel, click time (timestamp (UTC)), app downloads time and is_attributed (target attribute). Table 2 summarizes the acquired data.

Next, pre-processing module is developed to clean and convert the raw data into reliable format in order to make it readable for machines. In pre-processing, time attribute was removed during the pre-processing data stage. Click

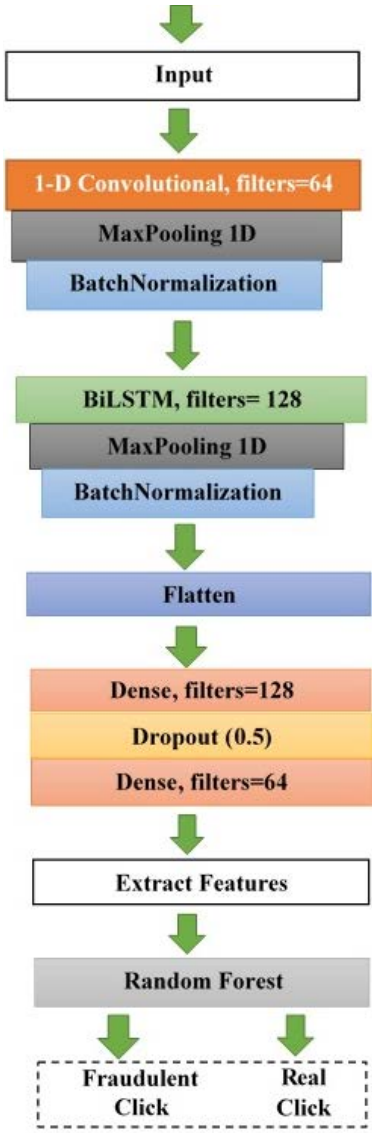


FIGURE 6. Flow model of the proposed CNN-BiLSTM-RF architecture.

TABLE 2. Dataset description.

#	Feature	Description
1	IP address	It indicates the click IP address.
2	App	It represents application identifier for marketing purposes.
3	Device	It indicates the click IP address.
4	OS	An identifier for the user's mobile phone's operating system version.
5	Channel	The publisher's channel id for mobile advertisements.
6	Click time	It shows click timestamp.
7	Downloads time	It represents download time for the downloaded application through clicking on an advertisement.
8	Is_attributed	It is the target attribute to be predicted.

Algorithm 1 Proposed CNN+BiLSTM-RF Algorithm

Input: Input clicks X Data, y is the target variable, F represents features and E_F represents extracted features.
Output: Detection of real and fake clicks in talking dataset.

$Encoding_X \leftarrow EncodingData$
$Temporal_F \leftarrow Features(X_{timestamp})$
$Combined_{data} \leftarrow Combined(X, Temporal_F)$
$Reduced_F \leftarrow CorrelationIndex(Combined_{data})$
Apply SMOTE to balance Samples Distribution per class labels as $Balanced_{samples}$
$Normalized_F \leftarrow Normalize(Balanced_{samples})$
Divide $Normalized_F$ into $K(K = 10)$ subsets as $[S_1, S_2, S_3, \ldots, S_K]$
for $X, y \in S_K$ **do**
  for $e \in (1, 100)$ **do**
    Train CNN-BiLSTM as $Hybrid_{model}$ using $f(X, y)$ Where $X, y \in [S_1, \ldots S_K]$
    Validate $Hybrid_{model}$ using $f(X, y)$
    Calculate Training and Validation Accuracy as $Training_{Acc}$ and $Validation_{Acc}$
    Calculate Training and Validation Loss as $Training_{val}$ and $Validation_{val}$
    Update Weights using $Adam_{optimizer}$ to reduce $Loss_{val}$ for $e + 1$
  end
  Report Training and Validation Accuracy for S_i as $Global_{Accuracy}^{Training}$ and $Global_{Accuracy}^{Validation}$
end
E_F from training Samples using trained $Hybrid_{model}$
Train RF_{model} using $f(E_F, y_{train})$
$\bar{E_F}$ from Unseen Samples using trained $Hybrid_{model}$
Test trained RF_{model} using $f(\bar{E_F}, y_{test})$
Obtain y_{pred} from $f(\bar{E_F}, y_{test})$
Evaluate RF_{model} results using $f(y_{test}, y_{pred})$ to compute Confusion Matrix CF_m
Compute Accuracy, Precision, Recall, F1 Score and AUC using CF_m

time attribute was divided into four sub-columns: day, hour, minute, and second. Next, label encoding is employed to convert the labels into a numerical form so that they can be converted into machine-readable form. Label encoding converts data into a form the computer can understand but assigns a unique number to each category of data. If the datasets are not well-organized, this could lead to problems with their training. A label with a high value may be given higher priority than a label with a lower value [63]. In this way, label encoding is used to convert the categorical values of the following attributes, such as Device, OS, and Channel into the numeric form.

For the experiments, due to limited computational resources, the entire dataset was not considered. Therefore, only 1 million data samples are considered, and the class ratio matches the ratio for 200 million samples. Dealing with unbalance datasets, the ML and DL algorithms more biased with dominant class [64]. Therefore, to address this problem, we used the Synthetic Minority Oversampling Technique (SMOTE) oversampling technique to balance the unbalanced dataset. SMOTE is a popular oversampling technique that turned into proposed to enhance random oversampling however its conduct on high-dimensional records has now no longer been very well investigated [65].

B. EXPLORATORY ANALYSIS

In this subsection, a visual way is carried out to analyze the clicks data through box plots and heat-map. the primary reason for the box plot in our article is to locate the common number of the dataset to examine how the data is dispersed between each sample we compare the respective median of each Box. Furthermore, box-plot analysis is widely used to measure five value summary, such as minimum, lower quartile of the median, the median, the upper quartile of the median, and maximum Clicks.and we compare the respective median of each box plot. we analyze to investigate PPC according to time interval groups in terms of Hourly Clicks (HC), Daily Clicks (DC) and Weekly Clicks (WC). It can be seen that the relationship between HC, DC, WC, and PPC varies because of the different structures of Clicks. Heat-maps are utilized in diverse sorts of analytic however are maximum usually used to reveal Visitor Clicks on particular web-pages or website templates. and its indicates display wherein Visitor have clicked on a page, how ways they've scrolled down a page or used to show the consequences of eye-monitoring tests. Click analytics are useful for web activity analysis, marketing, software testing, market research, and users productivity analysis.

Therefore, clicks data are analyzed based on the following temporal granularity, such as hourly, daily, and weekly clicks analysis. First, hourly clicks are visualized to analyze total hourly traffic on PPC websites, we create clicks frequency that displays on y-axis and hourly clicks on x-axis. In this example, we're looking at hourly trends for instance 350k visitors click on PPC ad per hour between 23 days. Fig. 7 shows hourly clicks analysis.

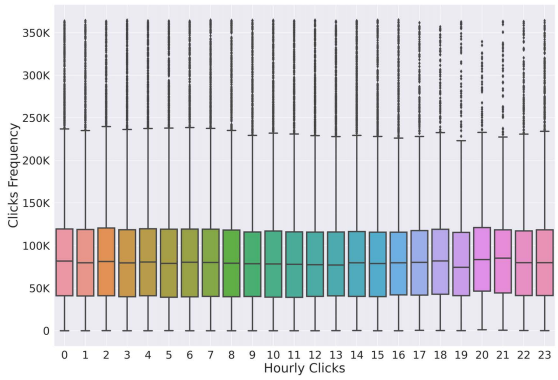


FIGURE 7. Positive and negative words analysis.

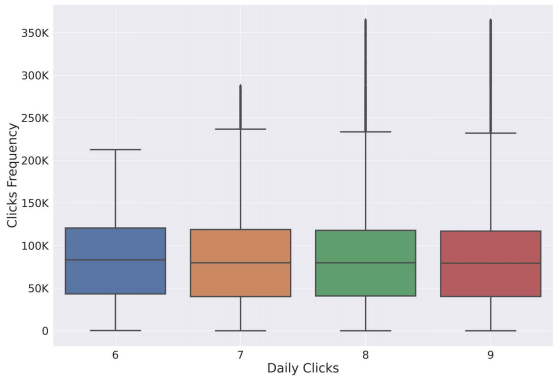


FIGURE 8. Daily analysis of clicks.

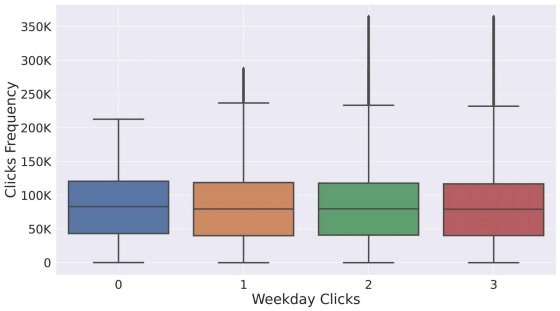


FIGURE 9. Weekly analysis of clicks.

Daily clicks analysis is a PPC insights aggregation of a user's tracked conduct throughout a website. in this graph we recognize daily clicks on a ad between 6 to 9 DC advanced analytic. Similarly, Fig. 8 shows daily clicks analysis.

Next, switch the graph to "Weekly Clicks Analysis" to look at data on an even more granular level.In weekly click analysis, we collect data from 3.5 million users from week one to week three. In comparison to the second week, fewer users clicked on the ad in the first week. We examined the clicking behavior of users on Pay Per Click Ads, which became more noticeable in the third week.

Fig. 9 illustrates weekly click analysis. It shows a weekly distribution of clicks data based on five value summary. The box plot analysis indicates that the week 2 and 3 achieved maximum number of clicks data.

Moreover, Fig. 10 shows correlation analysis to investigate the linear relationship between temporal and output features. Correlation analysis is a statistical technique used throughout analysis to ascertain the strength of a linear relationship between two variables and compute their association. temporal correlation analysis is a table that reveals the correlation coefficient between variables. Either every cell in the table represents the relationship between the two variables. A correlation matrix can also be used to summarize data, as an insight towards a more detailed analysis, or as a diagnosis and monitoring for advanced analyses. Our goal is to encapsulate a substantial number of data to determine patterns. In our preceding example, the perceptible pattern is that all of the variables are highly correlated with one another. The pairwise correlation analysis is considered to investigate the linear relationship between pairs of features. The pairwise correlation range varies between -1 to $+1$, where negative correlation indicates that the linear relationship between pairs of features is weak and positive correlation indicates that the linear relationship between features is strong.

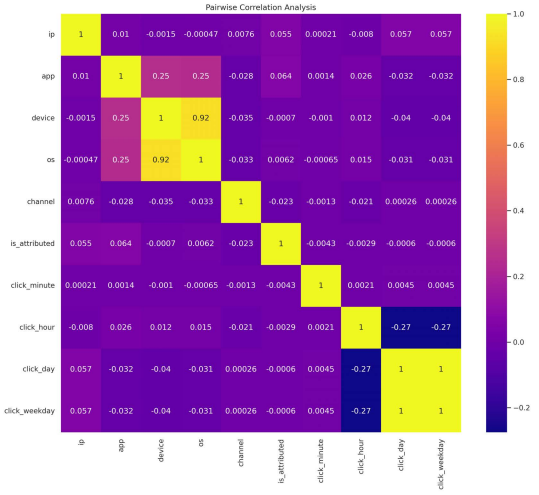


FIGURE 10. Correlation analysis of temporal patterns with respect to target attribute.

V. EXPERIMENTAL ENVIRONMENT, AND RESULTS ANALYSIS

A. IMPLEMENTATION ENVIRONMENT

The selected one million samples dataset are divided into 80% training data and 20% used for testing. The results are gained through 5-fold cross-validation techniques. For normalization, we used the Min-Mix scaler, while for balancing the dataset used SMOTE oversampling. First, we train the CNN-BiLSTM model on 100 epochs using the training data. The SGD optimizer is used for optimization, learning rate 0.001, batch size 33, and loss categorical cross-entropy. After training and validation, the CNN-BiLSTM model replaces the output layer, consisting of 2 filters, activation sigmoid and kernel regularizer l2 (le-4), with the RF classifier. The features are extracted from CNN-BiLSTM deep learning networks and feed into RF classifier for classification. The CNN extracts the deep feature, and BiLSTM can grip in a data sequence long-term dependency. Finally, we run the experiment for RF with the number of estimators 200 and random state 42. Table 3 presents a detailed summary of the implementation environment for our proposed architecture.

B. RESULTS ANALYSIS

This subsection investigates the performance of the proposed ensemble model. First, loss and accuracy of the DL models are evaluated using training and validation datasets.

TABLE 3. Implementation environment of the proposed CNN-BiLSTM-RF architecture.

Hardware	Description
Operating System	Microsoft Windows 10
Main Memory	32 GB
Accelerator	GPU
Programming Language	Python
IDE	PyCharm
Libraries	Pandas, Sklearn, Keras, TensorFlow, MatplotLib and Seaborn.
Persistence Storage	MS Excel (CSV File)

Second, performance of each implemented model is evaluated and compared using the evaluation indicators such as confusion matrix, accuracy, precision, recall, F1-score, and AUC.

1) PERFORMANCE ANALYSIS

The confusion matrices are shown in Fig. 11 to compare the performance of the proposed deep CNN-BiLSTM-RF and other implemented DL models, such as BiLSTM, CNN and CNN-BiLSTM architectures for click fraud detection. The confusion matrix is utilized to analyze the CNN-BiLSTM-RF model performance. We evaluated the model performance on 39910 testing samples (real: 19959 clicks and fraudulent: 19951 clicks). The dark green diagonal of the matrix represents the accurate classifications, whereas all other entries are mis-classifications. As illustrated in Fig. 11a, when BiLSTM is individually applied on testing data, 189 real clicks are inaccurately classified as fraudulent (false negative) and 1,778 fraudulent clicks are inaccurately classified as real (false positive). Similarly, mis-classification rate of CNN model for real and fraudulent clicks are 125 and 1417 as visualized in Fig. 11b, which indicates that CNN model performed well compared to the BiLSTM. However, when CNN-BiLSTM-RF is applied to the testing data, only 12 real clicks are mis-classified as fraudulent, whereas 154 fraudulent clicks are mis-classified as real as shown in Fig. 11d. Thus, the CNN-BiLSTM RF model performs significantly better than the CNN-BiLSTM and standalone BiLSTM and CNN models.

Furthermore, different evaluation measures are employed, such as accuracy, precision, sensitivity, specificity, F1-score and AUC to test and evaluate the results of the proposed model [66], [67]. Accuracy evaluates a predictor's ability to identify all instances correctly, either positive or negative as shown in equation 7:

$$Accuracy\,(ACC) = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{7}$$

Sensitivity is the frequency of accurately predicted positive samples among all true positive samples as follows in equation 8:

$$Sensitivity = Recall\,(RE) = \frac{T_p}{T_p + F_n} \tag{8}$$

Thus, it assesses the capacity of a predictor to identify positive samples. Similarly, specificity evaluates the capacity of a classifier to identify negative instances. Equation 9 shows a basic formula to estimate precision for a binary classification problem.

$$Specificity = Precision\,(PR) = \frac{T_p}{T_p + F_p} \tag{9}$$

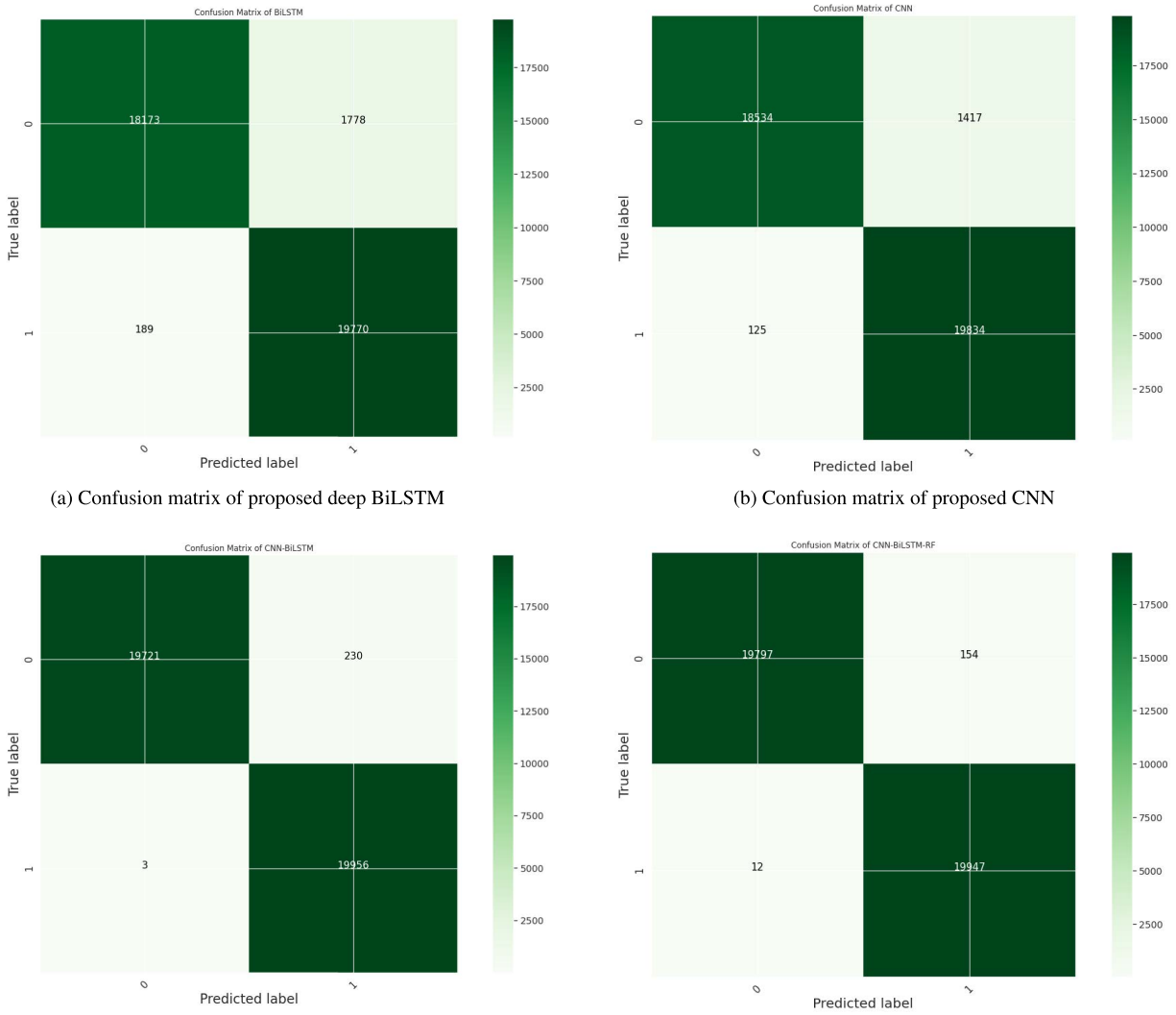
Furthermore, Harmonic mean of precision and recall is called F1-score. The formulas of measures are given below in equation 10:

$$F1Score = 2 * \frac{RE * PR}{RE + PR} \tag{10}$$

Based on evaluation analysis, Table 4 analyzes the click fraud detection performance of proposed CNN-BiLSTM and CNN-BiLSTM-RF models. Furthermore, Table 4 shows that when the CNN-BiLSTM is applied to test data, the classification accuracy is only 98.09 $\pm$ 0.13% (with 96.56 $\pm$ 0.17% sensitivity and 99.74 $\pm$ 0.04% specificity) for real and fake clicks. As explained earlier, we used CNN-BiLSTM for feature extraction and fed the extracted feature into the RF algorithm for classification. The experimental results of the CNN-BiLSTM in combination with RF yield 99.19% accuracy (with a sensitivity of 98.50 $\pm$ 0.11% and a specificity of 99.89 $\pm$ 0.03%), which is the best performing ratio. An improvement in recall and precision is also noted. The CNN-BiLSTM achieved (99.74 $\pm$ 0.04% precision and 98.12 $\pm$ 0.13 f1-score) and the CNN-BiLSTM-RF architecture obtained (99.89 $\pm$ 0.03% precision and 99.19 $\pm$ 0.08% f1-score). It shows that the CNN-BiLSTM model extracts meaningful features that help to enhance CNN- BiLSTM-RF performance. Table 4 shows the overall accuracy, precision, recall, f1-score and AUC of the proposed and other implemented models.

2) LOSS/ACCURACY ANALYSIS OF DL MODELS

Loss and accuracy of the implemented DL models are analyzed using the training and validation data samples sets. Loss indicates the error rate and defined as the summation of error to measures that how our proposed model is doing job well or bad. In this research study, categorical cross entropy is used as a loss function to estimate the loss for the given binary problem. It is used as loss function to get class labels in a one hot encoding format, such as 0's and 1's. Fig. 12 shows a loss analysis of the implemented individual and ensemble DL models. The training and validation loss of the proposed deep CNN-BiLSTM model is compared with BiLSTM, CNN and CNN-BiLSTM (with 1 layer for each). It can be analyzed that the training and validation loss of the proposed deep CNN-BiLSTM-RF model significantly decreases as the number of training epochs increases compared to the standalone models. The training and validation loss of the proposed deep CNN-BiLSTM model is varied between 0.01 and 0.35, which indicates that our proposed model is doing great job for detecting fraudulent clicks.



(a) Confusion matrix of proposed deep BiLSTM

(b) Confusion matrix of proposed CNN

(c) Confusion matrix of proposed CNN-BiLSTM

(d) Confusion matrix of proposed deep CNN-BiLSTM-RF

FIGURE 11. Comparison of confusion matrices of proposed deep CNN-BiLSTM-RF and DL models.

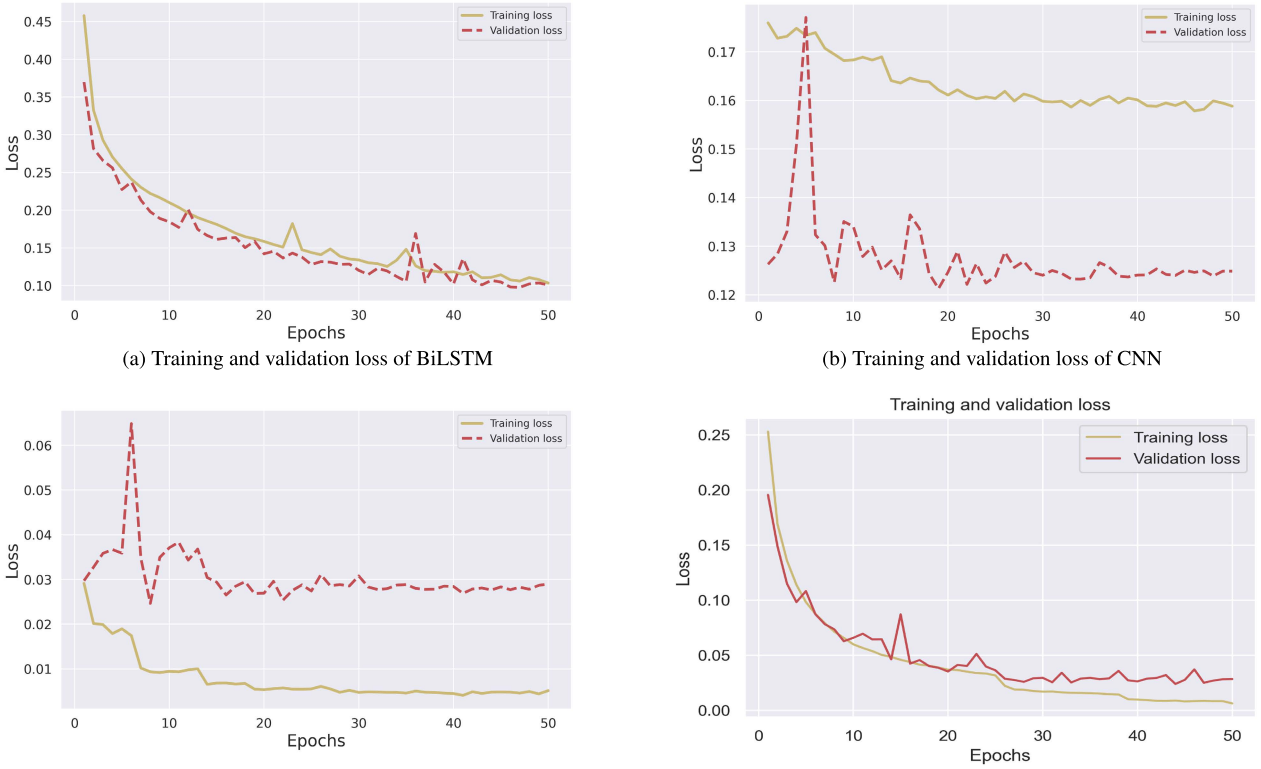
TABLE 4. The overall accuracy, precision, sensitivity, f1-score and specificity of the proposed model.

Model	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-score (%)	Specificity (%)
BiLSTM	96.27 ± 0.14	96.50 ± 0.19	96.30 ± 0.17	96.30 ± 0.21	96.27 ± 0.15
CNN	96.13 ± 0.17	96.30 ± 0.15	96.10 ± 0.19	96.10 ± 0.12	95.07 ± 0.17
CNN-BiLSTM	99.41 ± 0.13	99.40 ± 0.04	99.40 ± 0.17	99.40 ± 0.13	99.42 ± 0.12
CNN-BiLSTM-RF	**99.58 ± 0.08**	**99.60 ± 0.03**	**99.60 ± 0.11**	**99.60 ± 0.08**	**99.58 ± 0.08**

Similarly, accuracy is used as a performance metric to measure the performance of the model by comparing predicting and ground truth class labels. Furthermore, 50 epochs are used to calculate training and validating accuracy of each implemented DL model. Fig. 13 shows a comparison of categorical cross entropy based estimated accuracy of each individual and proposed ensemble models. The comparison shows a comparative analysis to analyze and compare the training and validation accuracy of the proposed deep CNN-BiLSTM-RF architecture with other implemented DL architectures. It is found that the training and validation accuracy of the proposed architecture increases as the number of training epoch increases. It can be seen from the comparative analysis that the accuracy of our proposed deep CNN-BiLSTM-RF model reached 99% for both training and validation sets as the epochs reached up to 50.

3) ROC CURVE ANALYSIS

The Receiver Operator Characteristic (ROC) curve is a binary classification problem performance measure. ROC curve provides a visual way to understand the trade-off between



(a) Training and validation loss of BiLSTM

(b) Training and validation loss of CNN

(c) Training and validation loss of CNN-BiLSTM

(d) Training and validation loss of proposed deep CNN-BiLSTM

FIGURE 12. Training and validation loss analysis of implemented standalone and proposed ensemble models.

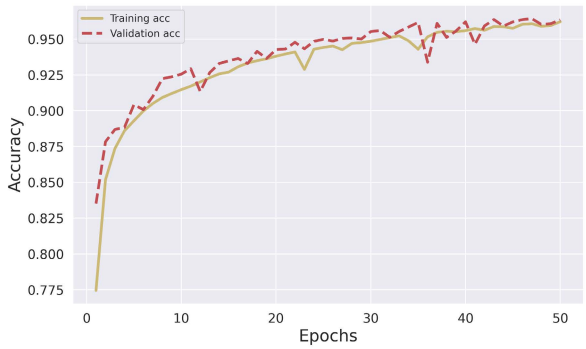
TABLE 5. Comparative analysis of the proposed CNN-BiLSTM-RF and existing click fraud detection models.

Study	Year	Approach	Dataset	Performance				
				ACC (%)	PR (%)	RE(%)	F1 Score (%)	AUC
[9]	2019	Autoencoder ANN- GAN	TalkingData	94.7	98	97	—	—
[28]	2019	Multi-model RF	Anti-Ad-fraud Data	—	91	83	—	95.20
[49]	2019	LightGBM	TalkingData	98	—	—	—	—
[32]	2020	XGBoost	——	91	91.5	91	—	—
[68]	2020	Cost-Sensitive CNN	BuzzCity Dataset	93	89	93	92	—
[46]	2020	Hybrid of graph embedding and DL	Mobile Advertising Data	—	—	—-	—-	93.5
[6]	2020	Random Forest	Real-Click Fraud Data	93.66	—	—	—	—
[5]	2021	Cascaded Forest and XGBoost	TalkingData, Avazu, kad	94.53	94	94	94	—
[43]	2021	XGBoost Gradient Boosting	Online Advertisement Data	96	—	—	—	—
[69]	2021	Hidden Markov Scoring Model	Mobile Advertising Company	94	—	—	—	—
Our	**2022**	**CNN-BiLSTM-RF**	**TalkingData**	**99.58**	**99.60**	**99.60**	**99.60**	**99.58**

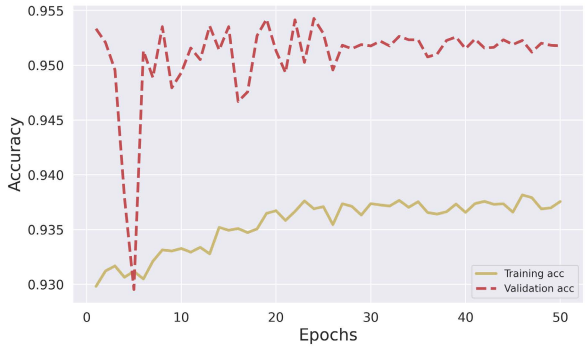
sensitivity (true positive rate) and specificity (false positive rate). It uses different probability thresholding values for error detection trade-off. It is effective and mostly used for balanced class distribution to investigate the rate of true and false positives. The higher value of y-axis shows that the performance of the proposed model is reliable and usually found perfect skill at a point (0,1). Fig. 14 illustrates the proposed model's performance. The given ROC curve analysis shows that the AUC of the proposed ensemble architecture is close to 1, which indicates that the true positives rate is higher than false negatives compared to the other models. The standalone models, such as BiLSTM and CNN achieved 96.27% and 95.07% AUC score, which indicates that CNN performance is slightly low compared to the BiLSTM. Similarly, CNN-BiLSTM achieved a 98.42% AUC score, while the CNN-BiLSTM-RF model obtained a 99.58% score. The ROC curve analysis indicates that our proposed deep CNN-BiLSTM-RF model outperformed the standalone and ensemble DL models.

VI. DISCUSSION

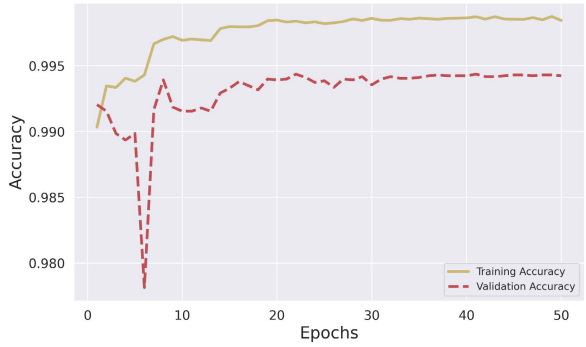
The experimental findings and analyses reveal that our proposed deep CNN-BiLSTM-RF performed well as compared to the CNN-BiLSTM architecture. The proposed deep CNN-BiLSTM-RF increased the accuracy of 3.31% and 3.45% compared to the standalone BiLSTM and CNN models as shown in Fig. 15. Similarly, it is also achieved a better accuracy compared to the ensemble model, such as



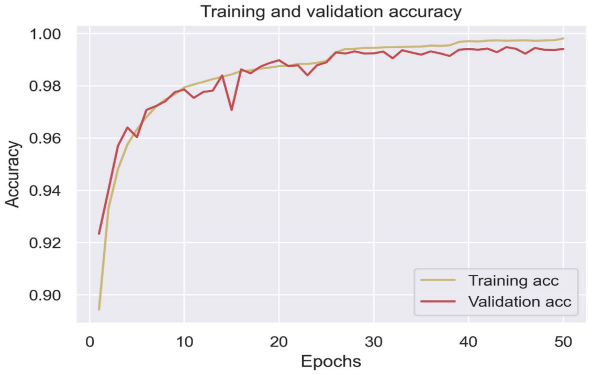
(a) Training and validation accuracy of BiLSTM



(b) Training and validation accuracy of CNN

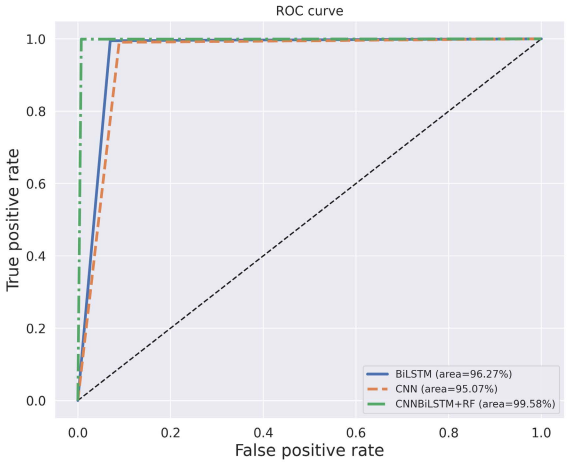


(c) Training and validation accuracy of CNN-BiLSTM

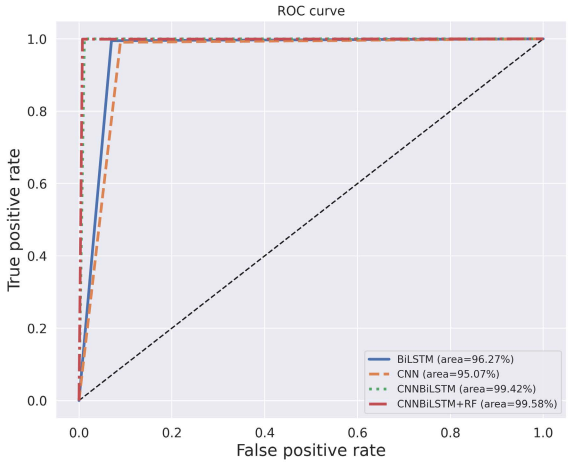


(d) Training and validation accuracy of proposed deep CNN-BiLSTM

FIGURE 13. **Training and validation accuracy analysis of implemented standalone and proposed ensemble models.**



(a) Proposed deep CNN-BiLSTM-RF and standalone DL models



(b) Proposed deep CNN-BiLSTM-RF and standalone and ensemble DL models

FIGURE 14. **ROC curves analysis of proposed deep CNN-BiLSTM-RF and other DL models.**

CNN-BiLSTM. The detection rate of the proposed model for fraudulent clicks detection is also improved by 3.1% and 3.3% compared to the BiLSTM and CNN models. In addition, our proposed model an improved f1-score by 3.3% and 3.5% compared to the BiLSTM and CNN models. Hence, our proposed deep CNN-BiLSTM-RF model achieved better performance compared to the BiLSTM, CNN and CNN-BiLSTM models.

Furthermore, Fig. 16 comparison of accuracy and precision (detection rate) for fraudulent clicks detection. The comparison indicates that our proposed model achieved high detection rate or fraudulent clicks detection of 99.60% compared to other listed models.

Moreover, Table 5 compared our proposed CNN-BiLSTM-RF model with some recent approaches applied for click fraud detection on the basis of accuracy, Recall, precision,

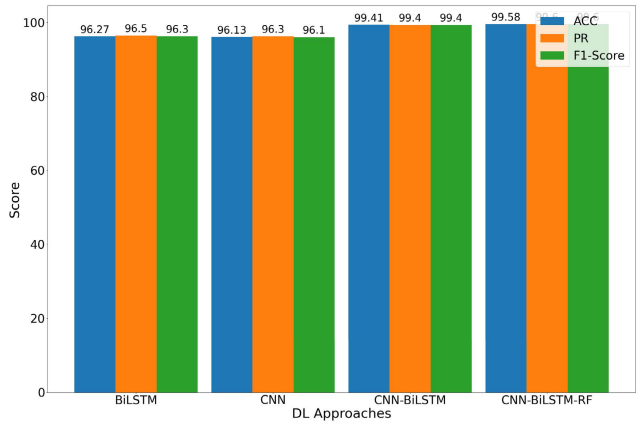


FIGURE 15. Comparative analysis of proposed and standalone models.

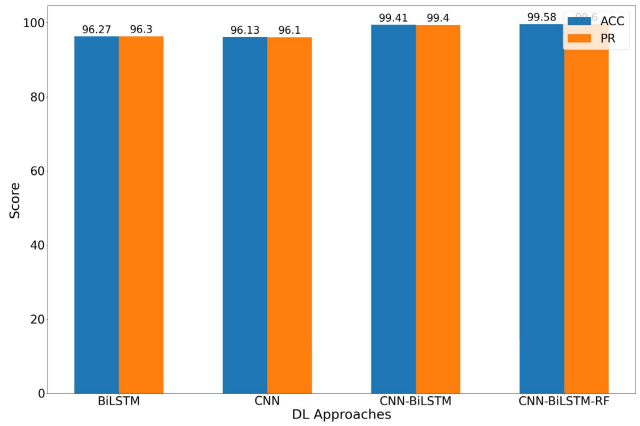


FIGURE 16. Comparison of accuracy and precision of proposed and standalone models.

F1-Score, specificity, and AUC. The research studies including [5], [6], [9], [32] and [68] achieved moderated accuracies between 91 and 94%. The studies, such as [49] and [43] achieved the best accuracies between 95 to 98%. Compared to our proposed model in terms of accuracy, precision, sensitivity, specificity and AUC, our proposed CNN-BiLSTM-RF achieved better results than themes.

VII. CONCLUSION AND FUTURE DIRECTION

Companies are changing their focus to advertising their items and services on mobile applications and websites as the online advertising market continuously grow. As a result, the problem of click fraud has become highly prevalent in recent years. Click fraud is the malicious or illegal clicking on adverts that results in the advertiser's revenue being squandered. To address this problem, numerous approaches have been proposed to detect click fraud. By categorizing clicks into invalid and valid, click fraud detection method can employ to shield the advertisers. We proposed a hybrid of CNN and BiLSTM with a combination of the RF classifier for click fraud detection. The combined CNN-BiLSTM-RF model gives the best results over the click fraud data. It gets asses from the CNN's capability of features extraction as well as the BiLSTM ability to acquire long-term bidirectional dependencies. Besides, RF is an ensemble machine learning model that is more suitable for classification than the traditional classifier associated with deep learning networks. The proposed models were trained on the TalkingData click fraud dataset one million samples. We compared Two deep learning models, CNN-BiLSTM and CNN-BiLSTM-RF, across different configurations and concluded through experimental results that the CNN-BiLSTM-RF model performs well with an accuracy of 99.19%. Although, this proposed architecture can be utilized as a general model to combat the click fraud in pay-per-click advertising to protect advertisers from fraudsters who generate clicks on their advertisements illegally. In future work, the proposed model will train on other click fraud datasets to evaluate its performance. we will also develop a tool to detect click fraud in real work internet and mobile advertising environments.

REFERENCES

[1] A. Dash and S. Pal, "Auto-detection of click-frauds using machine learning," *Int. J. Eng. Sci. Comput.*, vol. 10, pp. 27227–27235, Sep. 2020.

[2] Y. Xie, D. Jiang, X. Wang, and R. Xu, "Robust transfer integrated locally kernel embedding for click-through rate prediction," *Inf. Sci.*, vol. 491, pp. 190–203, Jul. 2019.

[3] M. Kantardzic, C. Walgampaya, and W. Emara, "Click fraud prevention in pay-per-click model: Learning through multi-model evidence fusion," in *Proc. Int. Conf. Mach. Web Intell.*, Oct. 2010, pp. 20–27.

[4] L. Pan, S. Mu, and Y. Wang, "User click fraud detection method based on Top-Rank-K frequent pattern mining," *Int. J. Modern Phys. B*, vol. 33, no. 15, Jun. 2019, Art. no. 1950150.

[5] T. G. Thejas, S. Dheeshjith, S. S. Iyengar, N. R. Sunitha, and P. Badrinath, "A hybrid and effective learning approach for click fraud detection," *Mach. Learn. Appl.*, vol. 3, Mar. 2021, Art. no. 100016.

[6] Z. Li and W. Jia, "The study on preventing click fraud in internet advertising," *J. Comput.*, vol. 31, no. 3, pp. 256–265, 2020.

[7] M. S. Iqbal, M. Zulkernine, F. Jaafar, and Y. Gu, "Protecting internet users from becoming victimized attackers of click-fraud: Protecting internet users from becoming victimized attackers of click-fraud," *J. Softw., Evol. Process*, vol. 30, no. 3, p. e1871, Mar. 2018.

[8] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, vol. 2, Mar. 2004, pp. 749–754.

[9] G. S. Thejas, K. G. Boroojeni, K. Chandna, I. Bhatia, S. S. Iyengar, and N. R. Sunitha, "Deep learning-based model to fight against ad click fraud," in *Proc. ACM Southeast Conf.*, Apr. 2019, pp. 176–181.

[10] H. Wang, C. Zhou, J. Wu, W. Dang, X. Zhu, and J. Wang, "Deep structure learning for fraud detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 567–576.

[11] J. Vanhoeyveld, D. Martens, and B. Peeters, "Value-added tax fraud detection with scalable anomaly detection techniques," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105895.

[12] Y. Alghofaili, A. Albattah, and M. A. Rassam, "A financial fraud detection model based on LSTM deep learning technique," *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 498–516, Oct. 2020.

[13] W. Min, W. Liang, H. Yin, Z. Wang, M. Li, and A. Lal, "Explainable deep behavioral sequence clustering for transaction fraud detection," 2021, *arXiv:2101.04285*.

[14] M. Arya and H. Sastry, "DEAL—Deep ensemble algorithm' framework for credit card fraud detection in real-time data stream with Google TensorFlow," *Smart Sci.*, vol. 8, no. 2, pp. 71–83, Apr. 2020.

[15] Y. Lucas, P.-E. Portier, L. Laporte, L. He-Guelton, O. Caelen, M. Granitzer, and S. Calabretto, "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs," *Future Gener. Comput. Syst.*, vol. 102, pp. 393–402, Jan. 2020.

[16] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102596.

[17] N. Iqbal, F. Jamil, S. Ahmad, and D. Kim, "A novel blockchain-based integrity and reliable veterinary clinic information management system using predictive analytics for provisioning of quality health services," *IEEE Access*, vol. 9, pp. 8069–8098, 2021.

[18] N. Iqbal, A. Rizwan, A. N. Khan, R. Ahmad, B. W. Kim, K. Kim, and D.-H. Kim, "Boreholes data analysis architecture based on clustering and prediction models for enhancing underground safety verification," *IEEE Access*, vol. 9, pp. 78428–78451, 2021.

[19] N. Iqbal, F. Jamil, S. Ahmad, and D. Kim, "Toward effective planning and management using predictive analytics based on rental book data of academic libraries," *IEEE Access*, vol. 8, pp. 81978–81996, 2020.

[20] S. Siddiqui, M. S. Faisal, S. Khurram, A. Irshad, M. Baz, H. Hamam, N. Iqbal, and M. Shafiq, "Quality prediction of wearable apps in the Google play store," *Intell. Autom. Soft Comput.*, vol. 32, no. 2, pp. 877–892, 2022.

[21] P. Shamsolmoali, D. K. Jain, M. Zareapoor, J. Yang, and M. A. Alam, "High-dimensional multimedia classification using deep CNN and extended residual units," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 23867–23882, Sep. 2019.

[22] K. Praanna, S. Sruthi, K. Kalyani, and A. S. Tejaswi, "A CNN-LSTM model for intrusion detection system from high dimensional data," *J. Inf. Comput. Sci.*, vol. 10, no. 3, pp. 1362–1370, 2020.

[23] S. Nagaraja and R. Shah, "Clicktok: Click fraud detection using traffic analysis," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, May 2019, pp. 105–116.

[24] M. Bathula. (2021). *BWorld Click Fraud Detection Approaches to Analyze the Ad Clicks Performed by Malicious Code*. Accessed: Jul. 20, 2022. [Online]. Available: https://doi.10.1088/1742-6596/2089/1/012077

[25] Y. Zhang, Y. Fu, Z. Wang, and L. Feng, "Fault detection based on modified kernel semi-supervised locally linear embedding," *IEEE Access*, vol. 6, pp. 479–487, 2018.

[26] C. Cao, Y. Gao, Y. Luo, M. Xia, W. Dong, C. Chen, and X. Liu, "AdSherlock: Efficient and deployable click fraud detection for mobile applications," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1285–1297, Apr. 2021.

[27] M. S. Iqbal, M. Zulkernine, F. Jaafar, and Y. Gu, "FCFraud: Fighting click-fraud from the user side," in *Proc. IEEE 17th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2016, pp. 157–164.

[28] F. Kanei, D. Chiba, K. Hato, and M. Akiyama, "Precise and robust detection of advertising fraud," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 776–785.

[29] F. Kanei, D. Chiba, K. Hato, K. Yoshioka, T. Matsumoto, and M. Akiyama, "Detecting and understanding online advertising fraud in the wild," *IEICE Trans. Inf. Syst.*, vol. 103, no. 7, pp. 1512–1523, 2020.

[30] P. S. Almeida and J. J. Gondim, "Click fraud detection and prevention system for ad networks," *J. Inf. Secur. Cryptogr.*, vol. 5, no. 1, pp. 27–39, 2018.

[31] S. Ghosh, "Identifying click baits using various machine learning and deep learning techniques," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 1235–1242, Jun. 2021.

[32] B. Viruthika, S. S. Das, E. M. Kumar, and D. Prabhu, "Detection of advertisement click fraud using machine learning," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 1–8, 2020.

[33] R. Espinosa, H. Ponce, and S. Gutiérrez, "Click-event sound detection in automotive industry using machine/deep learning," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107465.

[34] N. Gohil and A. D. Meniya, "A survey on online advertising and click fraud detection," Nayanaba Gohil Dept. Inf. Technol., Shantilal Shah Eng. College, Bhavnagar, GJ, India, Tech. Rep., 2020.

[35] R. Mouawi, M. Awad, A. Chehab, I. H. E. Hajj, and A. Kayssi, "Towards a machine learning approach for detecting click fraud in mobile advertizing," in *Proc. Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2018, pp. 88–92.

[36] D. Sisodia and D. S. Sisodia, "Data sampling strategies for click fraud detection using imbalanced user click data of online advertising: An empirical review," *IETE Tech. Rev.*, pp. 1–10, Apr. 2021.

[37] J.-A. Choi and K. Lim, "Identifying machine learning techniques for classification of target advertising," *ICT Exp.*, vol. 6, no. 3, pp. 175–180, Sep. 2020.

[38] D. Kesavan, "Advertisement click fraud detection," *Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol.*, vol. 7, no. 3, pp. 364–369, May 2021, Accessed: Jul. 20, 2022, doi: 10.32628/CSEIT217375.

[39] P. Raghavan and N. E. Gayar, "Fraud detection using machine learning and deep learning," in *Proc. Int. Conf. Comput. Intell. Knowl. Economy (ICCIKE)*, Dec. 2019, pp. 334–339.

[40] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *J. Netw. Comput. Appl.*, vol. 112, pp. 126–141, Jun. 2018.

[41] D. Sisodia and D. S. Sisodia, "Gradient boosting learning for fraudulent publisher detection in online advertising," *Data Technol. Appl.*, vol. 55, no. 2, pp. 216–232, Apr. 2021.

[42] S. Almahmoud, B. Hammo, B. Al-Shboul, and N. Obeid, "A hybrid approach for identifying non-human traffic in online digital advertising," *Multimedia Tools Appl.*, vol. 81, no. 2, pp. 1685–1718, Jan. 2022.

[43] N. P. Gohil and A. D. Meniya, "Click ad fraud detection using XGBoost gradient boosting algorithm," in *Proc. Int. Conf. Comput. Sci., Commun. Secur.* Cham, Switzerland: Springer, 2021, pp. 67–81.

[44] G. S. Thejas, J. Soni, K. G. Boroojeni, S. S. Iyengar, K. Srivastava, P. Badrinath, N. R. Sunitha, N. Prabakar, and H. Upadhyay, "A multi-time-scale time series analysis for click fraud forecasting using binary labeled imbalanced dataset," in *Proc. 4th Int. Conf. Comput. Syst. Inf. Technol. for Sustain. Solution (CSITSS)*, Dec. 2019, pp. 1–8.

[45] Y. Tian, T. Ma, and M. K. Khan, *Big Data and Security: First International Conference, ICBDS 2019, Nanjing, China, December 20–22, 2019, Revised Selected Papers*, vol. 1210, Berlin, Germany: Springer, 2020.

[46] J. Hu, T. Li, Y. Zhuang, S. Huang, and S. Dong, "GFD: A weighted heterogeneous graph embedding based approach for fraud detection in mobile advertising," *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Sep. 2020.

[47] I. Aberathne and C. Walgampaya, "Smart mobile bot detection through behavioral analysis," in *Advances in Data and Information Sciences*. Singapore: Springer, 2018, pp. 241–252.

[48] M. Fallah and S. Zarifzadeh, "Practical detection of click spams using efficient classification-based algorithms," *Int. J. Inf. Commun. Technol. Res.*, vol. 10, no. 2, pp. 63–71, 2018.

[49] E.-A. Minastireanu and G. Mesnita, "Light GBM machine learning algorithm to online click fraud detection," *J. Inf. Assurance Cybersecur.*, pp. 1–12, Apr. 2019.

[50] M. Bathula, R. C. Tanguturi, and S. R. Madala, "Click fraud detection approaches to analyze the ad clicks performed by malicious code," *J. Phys., Conf.*, vol. 2089, no. 1, 2021, Art. no. 012077.

[51] D. D. Sasikala, "Machine learning approach for click fraud detection," Ph.D. dissertation, 2021.

[52] E. N. Osegi and E. F. Jumbo, "Comparative analysis of credit card fraud detection in simulated annealing trained artificial neural network and hierarchical temporal memory," *Mach. Learn. Appl.*, vol. 6, Dec. 2021, Art. no. 100080.

[53] F. Zhu, C. Zhang, Z. Zheng, and S. A. Otaibi, "Click fraud detection of online advertising–LSH based tensor recovery mechanism," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9747–9754, Jul. 2022.

[54] S. Sadeghpour and N. Vlajic, "Click fraud in digital advertising: A comprehensive survey," *Computers*, vol. 10, no. 12, p. 164, Dec. 2021.

[55] D. Sisodia and D. S. Sisodia, "Feature space transformation of user-clicks and deep transfer learning framework for fraudulent publisher detection in online advertising," *Appl. Soft Comput.*, vol. 125, Aug. 2022, Art. no. 109142.

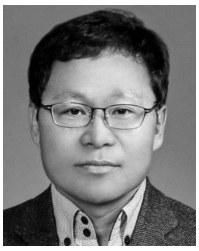
[56] S. Bhattacharjee, B. Saha, P. Bhattacharyya, and S. Saha, "Classification of obstructive and non-obstructive pulmonary diseases on the basis of spirometry using machine learning techniques," *J. Comput. Sci.*, vol. 63, Sep. 2022, Art. no. 101768.

[57] A. Akbarimajd, N. Hoertel, M. A. Hussain, A. A. Neshat, M. Marhamati, M. Bakhtoor, and M. Momeny, "Learning-to-augment incorporated noise-robust deep CNN for detection of COVID-19 in noisy X-ray images," *J. Comput. Sci.*, vol. 63, Sep. 2022, Art. no. 101763.

[58] M. Tzelepi and A. Tefas, "Class-specific discriminant regularization in real-time deep CNN models for binary classification problems," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1989–2005, Apr. 2020.

[59] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Proc. Comput. Sci.*, vol. 131, pp. 977–984, Jan. 2018.
[60] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2684–2691.
[61] M. Yang and J. Wang, "Adaptability of financial time series prediction based on BiLSTM," *Proc. Comput. Sci.*, vol. 199, pp. 18–25, Jan. 2022.
[62] *Talkingdata Adtracking Fraud Detection Challenge*. Accessed: Dec. 10, 2021. [Online]. Available: https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/
[63] H. Gupta and V. Asha, "Impact of encoding of high cardinality categorical data to solve prediction problems," *J. Comput. Theor. Nanoscience*, vol. 17, no. 9, pp. 4197–4201, Jul. 2020.
[64] M. Rahman, Y. Cao, X. Sun, B. Li, and Y. Hao, "Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray," *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107252.
[65] G. Huang, Q. Chen, and C. Deng, "A new click-through rates prediction model based on deep & cross network," *Algorithms*, vol. 13, no. 12, p. 342, Dec. 2020.
[66] N. Iqbal, R. Ahmad, F. Jamil, and D.-H. Kim, "Hybrid features prediction model of movie quality using multi-machine learning techniques for effective business resource planning," *J. Intell. Fuzzy Syst.*, vol. 40, no. 5, pp. 9361–9382, Apr. 2021.
[67] N. Ahmad, L. Han, K. Iqbal, R. Ahmad, M. A. Abid, and N. Iqbal, "SARM: Salah activities recognition model based on smartphone," *Electronics*, vol. 8, no. 8, p. 881, Aug. 2019.
[68] X. Liu, X. Zhang, and Q. Miao, "A click fraud detection scheme based on cost-sensitive CNN and feature matrix," in *Proc. Int. Conf. Big Data Secur.* Cham, Switzerland: Springer, 2019, pp. 65–79.
[69] A. Iroshan and W. Chamila, "Real time mobile ad investigator: An effective and novel approach for mobile click fraud detection," *Comput. Informat.*, vol. 40, no. 3, pp. 606–627, 2021.

AMREEN BATOOL received the bachelor's degree from the GC University of Pakistan, the M.C.S. degree from the Virtual University of Pakistan, and the M.S. degree in computer science and technology from Tiangong University, Tianjin, China, in 2021. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Jeju National University, Republic of Korea. She is serving as a Project Coordinator at EUT Global Ltd. Her main role is to coordinate with clients and field engineers to plan project delivery. Her research interests include machine learning, deep learning, and blockchain technology.



YUNG-CHEOL BYUN received the B.S. degree from Jeju National University, in 1993, and the M.S. and Ph.D. degrees from Yonsei University, in 1995 and 2001, respectively. He worked as a Special Lecturer at Samsung Electronics, from 2000 to 2001. From 2001 to 2003, he was a Senior Researcher at the Electronics and Telecommunications Research Institute (ETRI). He was promoted to join Jeju National University as an Assistant Professor, in 2003, where he is currently an Associate Professor with the Department of Computer Engineering. His research interests include AI machine learning, pattern recognition, blockchain and deep learning-based applications, big data and knowledge discovery, time series data analysis and prediction, image processing and medical applications, and recommendation systems.