Journal of Information Systems Engineering and Management
2025, 10(54s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Fake Profile Detection on Social Media Using Hybrid 2D CNN and AES-BiLSTM with Network Analysis

Mr. Sujit Kumar Badodia[1], Dr. Hemant Makwana[2]
[1](Devi AhilyaVishwavidyalaya),Information Techology, Institute of Engineering and Technology, Indore, India
[1]sbadodia@gmail.com
[2](Devi AhilyaVishwavidyalaya),Information Techology, Institute of Engineering and Technology, Indore, India
[2]hmakwana@ietdavv.edu.in

ARTICLEINFO

Received: 30 Dec 2024

Revised: 05 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

Social media platforms transform communication by enabling users worldwide to connect, share and interact effortlessly. With thousands of new users joining daily these platforms generate an immense volume of data including text, images and videos. While this growth fosters engagement and connectivity, it also presents challenges such as the proliferation of fake accounts and online impersonation. Malicious profiles often exploit these networks to distribute malware, viruses and harmful URLs, undermining the trust and security of social media ecosystems. Detecting and eliminating fake profiles is crucial to maintaining the integrity of these platforms and ensuring user safety.This paper proposed a hybrid artificial intelligence model to address these challenges by combining advanced preprocessing, feature extraction and classification techniques. During the preprocessing phase, the model removes duplicate entries, imputes missing values using mean imputation and scales the data with the Min-Max normalization technique to ensure consistency. For feature extraction, Principal Component Analysis (PCA) is used to decrease the dimensionality of the data, ensuring efficient processing. The hybrid classification framework integrates a 2D Convolutional Neural Network (2D CNN) to extract spatial features from the input data. Additionally, an Attention-Enhanced Stacked BiLSTM (AES-BiLSTM) is utilized to capture temporal features, while the added attention mechanisms improve the model's focus on the most relevant information.The 2D CNN and AES-BiLSTM models are further enhanced through hybrid optimization, with their hyperparameters fine-tuned using the Seagull Optimization Algorithm (SOA). Implemented in Python, this approach achieves a high accuracy of 98.9%, precision of 98.9%, specificity of 98.9%, and an F1-score of 99%. It provides a robust and scalable solution for detecting fake profiles on social media platforms. By addressing the limitations of traditional methods, this model offers an effective framework to safeguard social media ecosystems against fraudulent activities.

Keywords: PCA, 2D CNN, AES-BiLSTM, Fake profile detection, Seagull Optimization Algorithm (SOA).

INTRODUCTION

Social media platforms have revolutionized communication enabling users worldwide to connect, share and interact effortlessly [1]. With thousands of new users joining daily the volume of data generated is immense, encompassing text, images and videos [2]. While this growth facilitates engagement it has also introduced challenges such as fake accounts and online impersonation. These malicious profiles are often used to spread malware, viruses and harmful URLs, undermining the trust and security of social media platforms [3-5]. The ability to detect and eliminate fake profiles is essential to maintaining the integrity of these networks and safeguarding users.

The detection of fake profiles remains a significant challenge due to the complexity and evolving nature of fraudulent behavior on social media [6-9]. Traditional heuristic-based methods often fall short as they are unable to cope with the sheer scale of data and the sophisticated tactics employed by fraudsters [10,11]. Fake profiles often

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

exhibit subtle characteristics that make them difficult to identify and current approaches struggle with accurately classifying profiles particularly in large and diverse datasets [12]. There is a pressing need for advanced machine learning models capable of handling the vast data landscape identifying fake profiles in their early stages and providing scalable, accurate and efficient solutions to address this growing concern [13-15].To tackle these challenges we propose a hybrid artificial learning model that combines advanced preprocessing, feature extraction and classification strategies.

The key contributions are depicted bellow:

- Initially, the data is collected from CSV files containing information about online social media profiles, including user posts, comments and profile details.
- The preprocessing step focuses on transforming the raw data into a complete and standardized rating matrix by addressing missing values and ensuring consistency to facilitate effective model training.
- After preprocessing, features are extracted using the Principal Component Analysis (PCA), generating both 1D and 2D features.
- The extracted 2D and 1D features are processed by a 2D CNN and AES-BiLSTM,optimized using the Seagull Optimization Algorithm (SOA)
- The outputs from the 2DCNN and AES-BiLSTM models are concatenated and fused into a unified datasetand then hybrid model used to classify profiles as either fake or authentic.
- The proposed approach is implemented in Python and evaluated using various metrics including accuracy, sensitivity, specificity and F1-score.

LITERATURE SURVEY

Several investigators have established effective methods for fake profile detection on social media platforms, which can be analyzed as follows: .Kadam, N, and Sharma, S. K. et al.[16] discussed fake profile detection using a Twitter dataset to demonstrate an ML-based approach to fake news detection. The proposed model incorporates preprocessing to refine content and attributes, improving dataset quality and reducing data dimensionality. Five machine learning algorithms KNN, SVM, ANN, Naive Bayes and C4.5 decision trees were applied, with performance evaluated using a 4-fold cross-validation process. Metrics such as accuracy, error rate, memory usage, and time complexity were analyzed identifying two effective and accurate classification techniques. These techniques were used to develop an improved fake profile detection model, though challenges in computational complexity remained unresolved.

Ajesh, F. et al.[17] proposed a hybrid artificial learning model for detecting fake profiles demonstrating its effectiveness in achieving high accuracy. While traditional classification methods exist for recognizing fake accounts on social networks, this study introduced NLP techniques combined with machine learning algorithms to enhance detection. The approach utilized SVM, Random Forest, and Optimized Naïve Bayes algorithms, significantly improving detection accuracy. However, despite the advancements in accuracy, the method faced challenges in addressing computational complexity.

Amankeldin, D. et al.[18] introduced a deep neural network (DNN) method for social network fraud detection. The program is intended to acquire intricate features and patterns that distinguish between authentic and fraudulent profiles after being trained on a sizable dataset of both. Using a deep convolutional neural network technique created with 16 content-driven and profile-based characteristics, the study also sought to determine the minimal set of profile information needed for Facebook fake profile identification. The findings showed that the suggested approach was highly accurate in identifying phony profiles, underscoring its potential to improve the legitimacy and dependability of online social networks. Nevertheless, there are still issues with properly handling the intricacies of DNN-based fake profile detection

Sahoo, S. R., and Gupta, B. B. et al.[19] proposed a machine learning-based method for identifying dubious profiles with the goal of protecting Facebook's multimedia big data. Heterogeneity, human-centric content, and vast

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(54s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

amounts of text, audio, and video produced across online social networks are characteristics of this kind of dataset. When compared to alternative approaches, the experimental findings showed better performance when content-based and profile-based characteristics were used. However, the approach faced challenges related to reduced performance, lower resource efficiency and increased computational complexity.

Aditya, B. L. et al.[20] presented a sophisticated deep-transfer learning model that uses a thorough examination of several social media data samples to detect phony profiles. Posts, likes, comments, multimedia, user activity, and login behaviors are just a few of the platforms from which the model gathers data. It analyzes all kind of data to find odd trends that point to fraudulent identities, such male profiles who regularly post or use pictures of women. In order to minimize feature redundancy, feature optimization is carried out using the Elephant Herding Optimizer (EHO) for 1D data and the Grey Wolf Optimizer (GWO) for 2D data. The results demonstrated improvements in accuracy, precision and recall compared to traditional methods. However, the model faced limitations in achieving comprehensive performance analysis.

Wanda, P., and Jie, H. J et al.[21] discussed the growing popularity of Online Social Networks (OSNs) for sharing text, photos and videos while addressing the challenge of fake accounts, they are frequently used to propagate infections, malware, or fraudulent websites. They developed DeepProfile, a deep neural network (DNN) method, to address the problem of bogus accounts, taking inspiration from the success of deep learning in computer vision, specifically in feature extraction and representation. Instead of conventional machine learning, a dynamic CNN was developed to train a model for fake profile classification, incorporating a novel pooling layer to enhance neural network performance during training. Experimental results showed promising improvements in accuracy, though the method fell short of achieving optimal efficiency.

Masood, F. et al.[22] discussed a study of methods for identifying Twitter spammers, with a taxonomy that groups methods according to their capacity to identify false users, spam in hot topics, spam connected to URLs, and phony content. A variety of features, such as user, content, graph, structure, and time-related elements, were used to compare the methods. Researchers interested in the latest advancements in Twitter spam identification should find this study to be a helpful resource. However, it failed to achieve optimal effectiveness in detecting spammers.

Murugan, S. et al.[23] introduced a method that ensures user security and trust while providing an efficient and scalable solution for tackling fake profiles on social media platforms. This added layer of analysis strengthens the overall approach to combating fraudulent profiles and enhances the effectiveness of the detection process. The system utilizes Google Custom Search for reverse image searches on profile photos and a Naive Bayes classifier to assess profile completeness. However, continuous development and adaptation of the strategy are needed to address emerging challenges in identifying fraudulent accounts.

PROPOSED METHODOLOGY

This study focuses on detecting fake profiles on social media. CSV files are given as input and during preprocessing stage duplicate entries are removed, missing values are handled using mean imputation and the data is scaled using the Min-Max normalization technique to ensure consistency.For feature extraction, Principal Component Analysis (PCA) model is utilized to derive two types of features: 2D and 1D. The 2D features are processed through a 2D Convolutional Neural Network (2DCNN), while the 1D features are analyzed using an AES-BiLSTM model. The outputs from the 2DCNNand AES-BiLSTMmodels are optimized using the Seagull Optimization Algorithm (SOA). The results of the two hybrid models are then concatenated and fused to generate a unified dataset which is used to classify profiles as either fake or authentic. The overall structure can be shown below.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

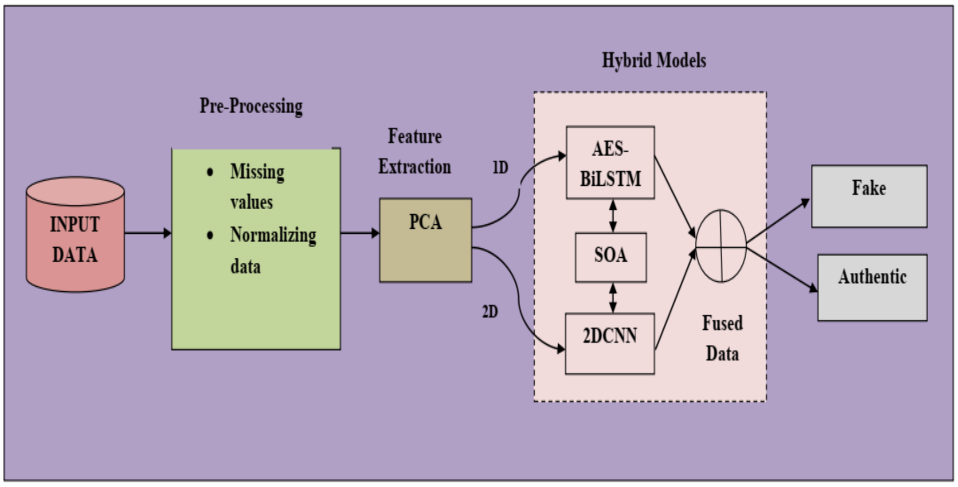

Figure 1: Overall Structure of Proposed Methodology

3.1 Pre-processing:

In the preprocessing stage, duplicate data is removed and missing values are addressed using mean imputation. Following this, the Min-Max normalization technique is applied to adjust data, ensuring it is clean, consistent and ready for input into the proposed model.

Consider a dataset D with N records each containing M attributes. The overall quantity of attribute values in the dataset is M×N. If any attributes in D have missing values, these entries are typically represented as non-numerical placeholders. To handle such missing values, this paper employs mean imputation where the associated attribute's means is used to substitute every value that is missing.

From the original dataset D, all rows containing missing values are removed, resulting in a new dataset d with n valid records.

The pseudo-code for the mean imputation process is as follows:

```
For c = 1 to M
Find mean value “Am” of all the attributes of the column ‘c’
Am(c) =(sum of all the elements of column c of d)/n
For r=1 to N
For c = 1 to M
If D(N,M) is not a Number (missing value), then Substitute Am(c) to D(N,M)
```

After imputing the missing values, the Min-Max normalization technique is used to reduce the attribute values, ensuring consistency across the dataset. The normalization process is applied using the following equation:

$$B_{new} = \frac{B_{current} - B_{\min}}{B_{\max} - B_{\min}} \qquad (1)$$

Where,

Bnew denotes normalized attribute

Bcurrent denotes current attribute

Bmin denotes minimum value attributes within the relevant column

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bmaxdenotes the maximum value attributes within the relevant column

3.2 Feature Extraction using PCA

After pre-processing, feature extraction follows with Principal Component Analysis (PCA) reducing dimensionality by transforming high-dimensional data into a space with less dimensions. PCA captures maximum variance through principal components, eliminating redundancy and improving analysis efficiency. A more detailed explanation of this process is provided below:

3.2.1 Principal Component Analysis (PCA)

The core concept of PCA is to linearly transform data into a lower-dimensional subspace by maximizing the variance of the data. This process results in an uncorrelated orthogonal basis set, where the principal components are orthogonal as they correspond to the eigenvectors of the symmetric covariance matrix.

Mathematically, for a dataset with k observations each observation is represented as an n-dimensional vector excluding the class label. Let

The steps involved in computing PCA are as follows:

Compute the m-dimensional mean vector μ using:

$$\mu = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{2}$$

Calculate the estimated covariance matrix S for the observed data using:

$$S = \frac{1}{k}\sum_{i=1}^{k} (x_i - \mu)(x_i - \mu)^t \tag{3}$$

Determine the eigenvalues and their corresponding eigenvectors of S, where . $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 0$

Derive k principal components from the k original variables using:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k \tag{4}$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k \tag{5}$$

$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k \tag{6}$$

The y_k values are uncorrelated (orthogonal), with y1 capturing the maximum variance in the dataset, y2 accounting for the next highest variance and so on. Typically, in most practical datasets, a few larger eigenvalues dominate representing the majority of the variance.

$$\gamma_k = \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_m}{\lambda_1 + \lambda_2 + \ldots + \lambda_m + \ldots + \lambda_k} \geq 80\% \tag{7}$$

Here,γ_k represents the percentage of variance retained in the data representation. In PCA-based feature extraction, it is important to retain the principal components that explain at least 80% of the total variance. After this the features get extracted from the PCA model.

PCA can extract 1D and 2D features by projecting the original data onto its principal components. For 1D feature extraction, PCA reduces the data to a single dimension by selecting the dataset's largest variable is captured by its initial main component. The data is then projected onto this component, resulting in a 1D representation of the

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

original data. For 2D feature extraction, PCA identifies the top two principal components, each capturing a significant portion of the variance. The data is subsequently projected onto these two components, creating a 2D representation that highlights the main patterns or relationships within the dataset.

3.3 Hybrid Model using 2DCNN and AES-BiLSTM

In this section, a hybrid deep learning model is introduced, combining features of 2D-CNN and 1D-BiLSTM. This hybrid approach outperforms individual models by integrating the 2D-CNN and 1D-BiLSTM in parallel to complement each other's strengths. The 2D-CNN processes a 2D matrix, extracting key spatial features and patterns from the input data. A 2D structure allows the method to capture local relationships and dependencies between data points, such as in images or text, where spatial relationships are crucial. At the same time, 1D word embeddings or sequential data are input into the 1D-BiLSTM. This pattern is processed by the Bi-LSTM either forward as well as backwards, capturing temporal dependencies and contextual relationships between elements. This enables the model to learn the broader context of the sequence and how elements relate over time, enhancing its understanding of the input data. By combining the 2D-CNN for spatial feature extraction and the 1D-BiLSTM for temporal feature extraction, a hybrid model improves performance in tasks such as classification. A more detailed explanation of how the hybrid model functions is provided below:

3.3.1 2D CNN for extracting spatial features

CNN was created to automatically separate non-linear structures and sophisticated feature representations using highly variable data, particularly in computational vision along with processing images. In a context of fake profile detection on social media, we apply 2D-CNN to analyze data, such as user posts, comments, and interactions. The main objective of the 2D-CNN in this case is to capture hidden representations and identify potential features from the data, which can reveal patterns indicative of fake profiles, such as unusual posting behavior or irregular interaction patterns.Social media data is typically provided in raw 1D form, with each post or interaction recorded separately. However, 1D data lacks the broader contextual relationships and associations needed to identify fake behaviors. To address this, we transform the 1D data into 2D matrices. This 2D structure allows the 2D-CNN to perform convolution and pooling operations, capturing latent trends and hidden fluctuations in the data.In convolution operations, various filters are applied to learn feature representations from the data and generate feature maps. These feature maps help identify important characteristics, such as repeated phrases, unusual language patterns or the frequency of specific words that may be indicative of fake profile activities. After convolution, pooling operationsminimize features maps' spatial dimensions to increase model efficiency and concentrate on the more significant trends. Specifically, a max pooling strategy is used, where the highest values in each feature map's receptive field are selected, discarding a rest.

To avoid over fitting, were layers for dropout with normalized batcheslayers are included to prevent internal covariate shifts (ICS). Additionally, since the deep learning methods are extremely data-sensitive variability, it is essential to normalize the data before passing it to subsequent layers. This ensures that the model performs reliably and is less prone to issues such as gradient explosion or over fitting helping it better detect fake profiles based on data.The convolutional layer of the 2D-CNN is expressed mathematically as follows.

$$y^i = \sigma_i \left(w_i * x_i + b_i \right) \tag{8}$$

In this context, σi represents the sigmoid activation function, and yidenotes the output of the i-th convolutional layer. X_i represent the input data. The weight of the i-th convolutional layer is denoted by wi, while bi represents the bias term. After applying the convolutional operations, the output yi contains the feature maps that highlight important patterns within the data. Following this, pooling operations are conducted using a max pooling technique to lower the features maps' dimension. The max pooling layer formula is shown below.

$$y^m = \max_{i,j \in R\left(y^{ij}\right)} \tag{9}$$

The outcomes from the maximum pooling layers, that include the decreased map features following pooling processes, are denoted by ym in this case. A particular convolutional layer's j-th neuron is represented by its index,

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

j. Dropout and batch normalization layers are added to the model to avoid overfitting and address problems such as internal covariate shift (ICS). Furthermore, the 2D feature maps are converted into a 1D vector using a flatten layer, which connects the pooling layers. The spatial features are then recovered from the flatten layer using the following arithmetic equation:

$$y^f = g_i\left(w_i^f * y^m + b_i^f\right) \tag{10}$$

In this context, gi denotes the activation function, while along with correspond to a weight and bias terms respectively.

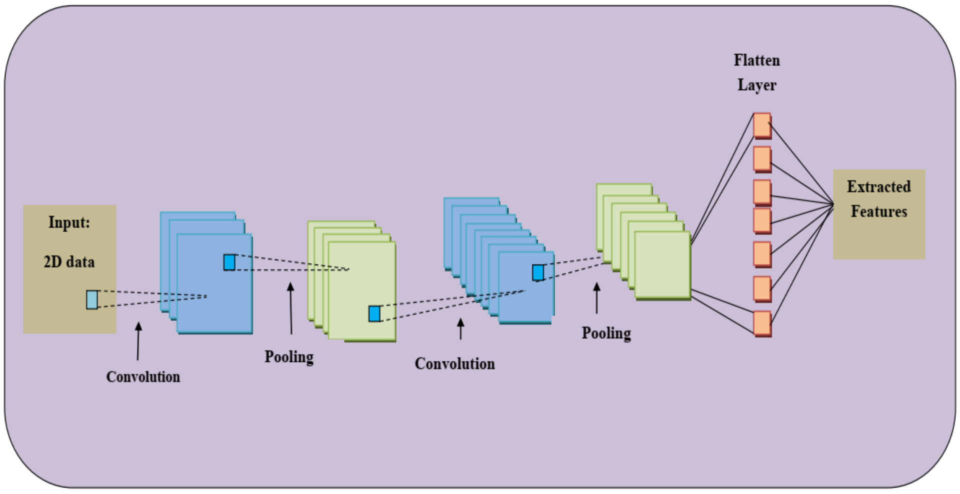

Figure 2: Architecture of 2D CNN

3.3.2 AES-BiLSTM for extracting temporal features

In this section, we utilize AES-BiLSTM to extract 1D features from the PCM model. A stacked BiLSTM, formed by combining multiple BiLSTM layers is employed for feature extraction. Subsequently the attention mechanism is used to enhance a feature representation. The detailed structure of AES-BiLSTM is presented below.

Hochreiter as well as Schmidhuber initially created LSTM networks, which are intended for effectively collecting dependence over time in data that is sequential. Three control gates that govern a memory cell activation vector c are the fundamental components of an LSTM, given an input pattern vector x=(x_1,x_2,...,x_n), where n is the initial length of the original input series.

The amount of the prior cell state ct-1 that is kept in the present cell state ct is decided by the gate that forgets. The gate that receives input regulates how much of the cell state c_t is updated by the present input x_t. Lastly, the output gate determines how far the output ht is influenced by the current cell state ct. Every gate is built as a fully connected layer that generates an output value in the [0,1] range after accepting a vector of values as input.

LSTM functionality is mathematically represented as follows:

Input gates: $$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + b_i\right) \tag{11}$$

Forget gates: $$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \tag{12}$$

Output gates: $$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \tag{13}$$

Cell states: $$c_t = f * c_{t-1} + i_t * \tanh.\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \tag{14}$$

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cell output: $$h_t = o_t * \tanh(c_t) \tag{15}$$

In this case, σ stands for the logistic sigmoid function. The given input sequence's t-th word vector is represented by x_t, while the state that is hidden is indicated by h_t. The b terms are the bias vectors (e.g., b_i for the input gate) connected across the three control gates and the W terms are the weight matrices (e.g., W_xf for the forget gate).

A stacked Bi-LSTM is presented to overcome the drawback of a single LSTM cell, which is limited to capturing the past context but not the future context. The output layer can use data from both past and future contexts thanks to its architecture, which includes two independent LSTM layers working in opposing directions.In a BiLSTM, the input sequence x=(x_1,x_2,...,x_n) is processed in both directions producing a forward hidden sequence and a backward hidden sequence $\vec{h}_t = (\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_n)$. The results of the both forward and backward sequences are concatenated to create the last encoded vector.

$$y_t = [\vec{h}_t, \overleftarrow{h}_t] \tag{16}$$

Previous studies have demonstrated that stacking multiple BiLSTM layers within a neural network can significantly enhance the performance of classification or regression tasks. Furthermore, theoretical evidence supports the notion that deep hierarchical models are more effective at representing certain functions compared to shallow ones. In our design, we define a stacked BiLSTM network where the output y_t from a lower layer serves as the input data for the upper layer that follows. Figure 3 shows the framework of the stacked BiLSTM.

$$h_t = W_{hh} \cdot \vec{h}_t + W_{hh} \cdot \overleftarrow{h}_t + b_h \tag{17}$$

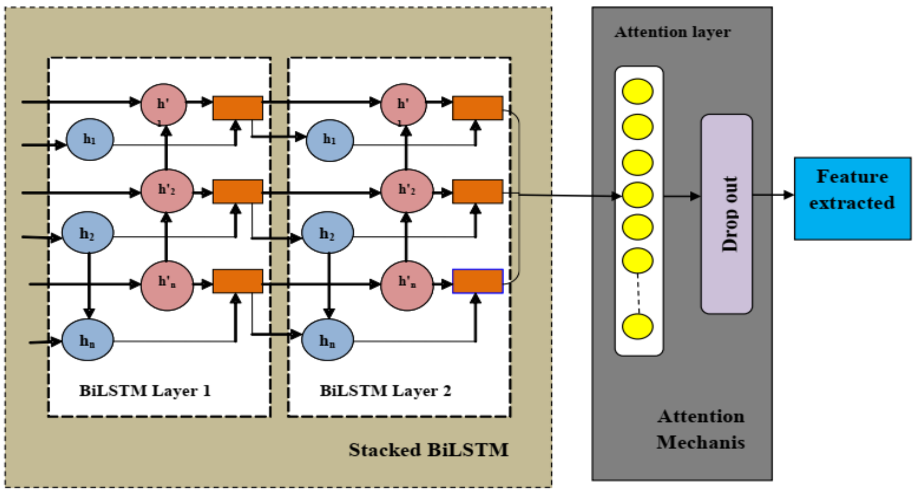

Figure 3: Architecture of AES-BiLSTM

The attention mechanism has emerged as a pivotal concept in diverse applications such as natural language processing and image recognition. Drawing inspiration from the human brain's selective visual processing, it prioritizes essential information while discarding irrelevant data to optimize task-specific outcomes. Its proven effectiveness in improving information discrimination has been widely documented in extensive research. Accordingly, this mechanism is seamlessly integrated into the proposed model framework, with its extracted temporal features is mathematical described as follows:

$$v_t = \tanh(\omega_p H_t + b_p) \tag{18}$$

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

$$\alpha_t = \frac{\exp\left((v_t)^T v_p\right)}{\sum_t \exp\left((v_t)^T v_p\right)} \tag{19}$$

$$z = \sum_t \alpha_t H_t \tag{20}$$

Whereas, z denotes the features extracted from the AES-BiLSTM model, denotes a weight matrix and b_p represent the bias vector. α_t is used to evaluate the similarity between v_t and the context vector v_p.

3.3.3 Seagull Optimization Algorithm (SOA) used for 2D CNN and AES-BiLSTM

In this section, the features extracted from both the 2D CNN and AES-BiLSTM are optimized using the SOA algorithm.The SOA generally simulates the migratory behavior of seagulls through a series of defined stages. Migration behavior models the swarming process of seagulls circling from one place to another, guided by three key conditions: avoiding collisions with neighbouring seagulls, moving toward a more favorable neighbouring guidance, and staying nearly the best-performing search engine. In order to avoid interactions in neighbouring seagulls, the parameteris utilized to improve what was seen value of the seagull's position, ensuring effective coordination during the optimization process.

Step 1: Initialization

In any optimization technique initialization is a crucial step. In this study the hyperparameters in the 2DCNN and AES-BiLSTM model are optimally selected. First the 2D CNN process involves initializing the seagull optimization (SOA) Algorithm with hyper parameters like the rate of learning (l), batch size (b), epochs (e) and regularization (r). Second the AES-BiLSTM process involves initializing the seagull optimization (SOA) Algorithm with hyper parameters like the dropout rate (d), batch size (s), activation function (a) and weight initialization (w). These parameters are randomly initialized and the initial population format is represented by the following equation (21)

$$x_i = \{x_1, x_2 \dots x_N\} \tag{21}$$

Here, represent the position of the N th solution it is expressed in next equation

$$x_N = \{l, b, e, r /_{2DCNN} ; d, s, a, w /_{AES-BiLSTM}\} \tag{22}$$

Equation 22 represents the 2D CNN parameters and the AES-BiLSTM parameters.

Step 2: Fitness calculation

Following the start-up procedure, fitness solutions are computed, with classification accuracy used as the fitness criterion. The fitness formula is provided in the following equation (23).

$$fitness = \max imum(accuracy) \tag{23}$$

Step 3: Update the Solution

This stage involves updating the agent's position based on the location of the best-performing agent following the given equation.

$$D_e = \rightarrow \left|\vec{P}_N + \vec{d}_e\right| \tag{24}$$

Here, De represents the separation between the agent and the agent with the highest performance. In the course of the migration, seagulls adjust their angles and rate of strike dynamically as they strike. They balance the weight of their bodies and wings to stay aloft. The atmosphere around it spirals as a result from this attacker motion, and its movement behavior over the x, y, and z surfaces can be characterized mathematically as follows.

$$\hat{x} = s \times \cos(g) \tag{25}$$

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

$$\hat{y} = s \times \sin(g) \tag{26}$$

$$\hat{Z} = s \times g, s = \alpha \times e^{\beta t} \tag{27}$$

In this case, g is an arbitrary number between the range of values and s represents the diameter that the spirals turn. The spiral's shape is determined by its parameters α and β, whereas e represents the fundamental logarithm. Based on these parameters the seagull's position is updated using the following formulation.

$$\vec{P}_C(i) = \left(\vec{D}_e \rightarrow + \hat{x} + \hat{y} + \hat{z}\right) + \vec{P}_b(i) \tag{28}$$

Represent the attack position of seagull.

Step 4: Termination

The algorithm iterates until a termination condition is reached, including the optimum number of repetitions, obtaining the target level of fitness or observing no significant improvement. Once terminated, it outputs the best solution and its fitness value. This approach is effective for optimizing hyperparameters, ensuring improved model performance.

Implementation steps are shown in pseudo-code in table 1

Table 1: Pseudocode of SOA

```
Algorithm 1:Pseudocode for ISOA

Input: hyper-parameters of 2DCNN and AES-BiLSTM ,
Output : Optimized  hyper-parameters
Initialize population
Set $f_c$ to 2;
Set u and v to 1;
While t<T
For i=1:N
Calculate seagull migration position $\vec{D}_s$ by equation (24)
Compute x',y'z',r using equation (25-27)
Calculate seagull attack position $\vec{P}_{s1}(t)$ by equation (28)
Update seagull optimal position $\vec{P}_{bs}(t)$;
t=t+1;
end for;
end while;
Output:  the best solution obtained by SOA
End SOA
```

After the optimization process, the features extracted from both models are combined to form a single fused dataset, which is then used to classify the results.

3.3.4 Classification using hybrid model

In this section, the features extracted from the 2D-CNN and AES-BiLSTM models are combined to form single fused features, which are then used to determine whether a user profile is fake or authentic. The details of this process are explained below:

Combining the Bi-LSTM and 2D-CNN models for processing data follows a structured approach. Initially, the 2D-CNN processes 2D representations of the input data, such as word or character embeddings extracting spatial

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

features and identifying patterns within the data. This step allows the model to capture relationships between adjacent data points highlighting important spatial characteristics. Simultaneously, the Bi-LSTM processes a sequence of information that records relationships over time and temporal dependency within the sequence. By considering both the forward and backward context, the Bi-LSTM enhances the capacity of the structureto understand a sequential nature on the input. Each model 2D-CNN and Bi-LSTM generates its respective output, the 2D-CNN produces y_f, and the Bi-LSTM yields z. These outputs were concatenated to form a combined feature vector, which is represented in a following equation

$$f_k = [y^f ; z] \tag{29}$$

In this context, [;] represents the concatenation operation used to merge feature vectors extracted from the 2D-CNN and 1D AES-BiLSTM into a single fused vector, fk which encapsulates both spatial and temporal information. This fusion enables the model to effectively leverage the complementary strengths of spatial features from the 2D CNN and temporal features from sequential data. Alternatively, a weighted sum approach can be employed for fusion, where specific weights are assigned to the spatial and temporal components, allowing more flexible and adaptive integration based on their relative importance for the task.

$$f_k = \alpha y^f + \beta z \tag{30}$$

In this approach, α and β represent the weights that determine the contributions of spatial and temporal features, respectively, with the constraint that α+β=1. This formulation allows the model to dynamically adjust the importance of each feature type, adjusting to the unique features and circumstances of the incoming data to optimize overall performance.

After fusing the features, the fused vector fk is passed to the hybrid module for classification, which predicts whether profiles are fake or authentic. The working process of the hybrid module is explained below:

The hybrid module creates a single feature vector by combining the outcomes from the Bi-LSTM and 2D-CNN modules. To make hybrids learning of both approaches easier, a joint weight matrix is constructed. Lastly, the merged feature vector is subjected to a sigmoid function in order to categorize the user profiles as either fake or authentic.

$$PRO_{\text{det}} = \sigma_h\left(W\left[y^f, z\right] + b\right) \tag{31}$$

Whereas,PRO_{det} denotes the combined output of 2D-CNN and AES-BiLSTM, represent the sigmoid activation function, y_f and zdenotes the final output of 2D-CNN and AES-BiLSTM models, W represents the joint weight for a hybrid model and b denotes the bias factor. The integration of features from both models enables the hybrid model to effectively classify the extracted features, thereby improving the accuracy of data detection. By harnessing the strengths of either temporally and spatially feature extraction, the hybrid model enhances the overall development for the classification task, making it a robust solution for distinguishing between fake and authentic profiles.

RESULTS AND DISCUSSIONS

4. Result and Discussions

The performance of the proposed PCA model for online social networking fraudulent account detecting was evaluated on a Python platform running in a system with an Intel Core i5 processor, 6GB of memory and Windows 10. The data used for evaluation was collected from online social media profiles, including user posts, comments, and profile details. The model predicts and classifies profiles as either fake or authentic based on the extracted features, demonstrating its ability to accurately distinguish between legitimate and fraudulent profiles.

4.1 Data description

A CSV file dataset: https://www.kaggle.com/datasets/debanshupal/fake-profile-detection-dataset,is a tabulated dataset in which every column denotes a certain attribute and each row is a data record. The first row contains column names, indicating the attributes. Columns can include various data types like integers, floats, strings, or

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

dates. For example, a CSV might have ID (integer), Name (string), Measurement (float, e.g., in cm) and Timestamp (datetime). The CSV files are easy to import into tools like Excel or Python for analysis and are commonly used for data from surveys, experiments or repositories.

4.2 Experimental results based on Fake profile detection

The following section presents an experimental analysis of fake profile detection based on accuracy, sensitivity, specificity, and F1-score. A detailed description of each evaluation metric is provided below:

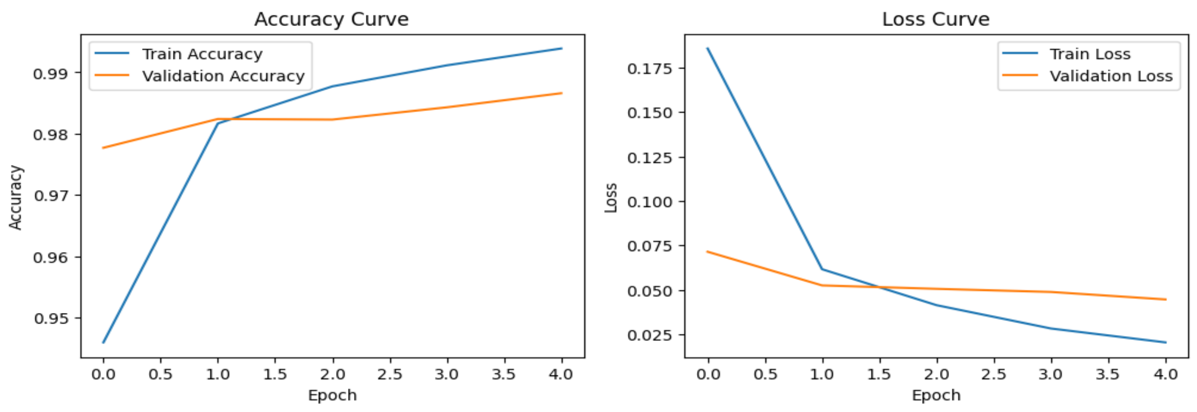

Figure 4: Performance based on accuracy and loss

Image 4 illustrates the instruction and verification performance of the classification model over 100 epochs, measured by accuracy and loss. In the first graph, representing Training & Validation Accuracy, the training accuracy (blue line) steadily increases, stabilizing around 0.97, which indicates effective learning using data that was used for training. But the correctness of the verification (red line) fluctuates between 0.92 and 0.94, showing a discrepancy that suggests potential overfitting, as the validation accuracy does not improve in parallel with the training accuracy. In the second graph, depicting Training & Validation Loss, the training loss (blue line) gradually decreases, signifying a reduction in error on the training data. Conversely, the validation loss (red line) remains relatively higher and continues to fluctuate, indicating overfitting, as the model's generalization based on verification information is poor.

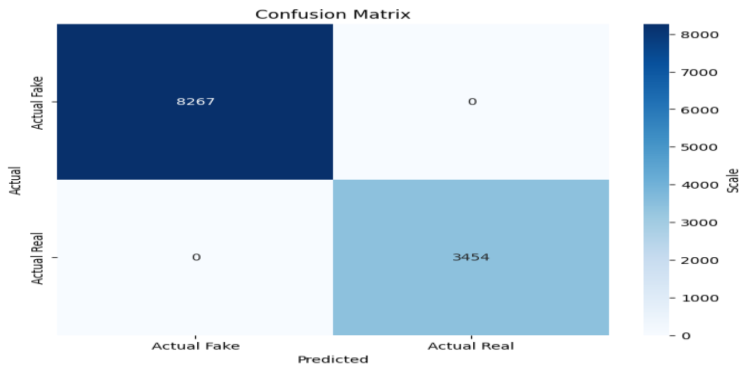

Figure 5: Confusion matrix

The effectiveness of classification of the suggested method is depicted in Figure 5 confusion matrix. The predicted values are displayed along the x-axis, whereas the y-axis displays the real outcomes. The two classes are labeled as 0 and 1. The matrix is presented in percentages, offering an efficient way to evaluate the classifier's performance. The high percentages in the true positive and true negative quadrants (3799 and 3834, respectively) indicate strong classification accuracy, while the relatively low percentages in the false positive and false negative quadrants (29 and 37) suggest that the model makes very few misclassifications.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

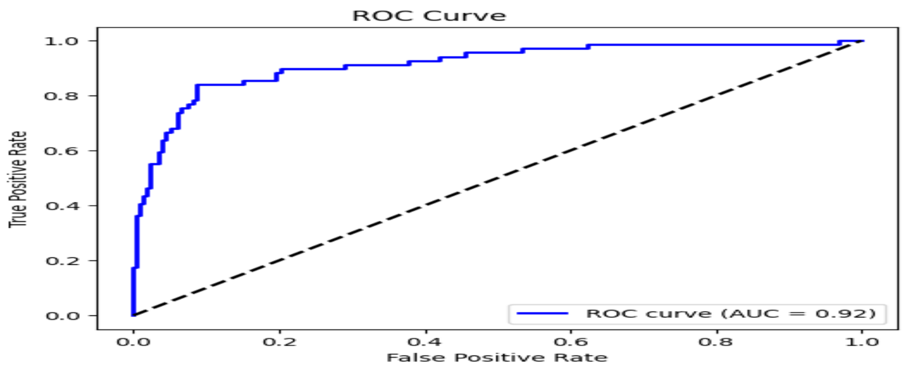

Figure 6: ROC Curve

A graphical instrument for assessing the binary classification performance, the Receiver Operating Characteristic (ROC) curve (Figure 6) illustrates the balance between the true positive rate (TPR) and the false positive rate (FPR) at different sensitivity levels. The y-axis in this plot displays the true positive rate (TPR), also known as sensitivity, indicating how well the model detects positive cases. The x-axis displays the false positive rate (FPR), which shows the proportion of negative instances that are mistakenly forecasted as positive. The blue curve shows the classifier's efficiency; a closer arc to the upper-left corner denotes a higher class differentiation. High predictive capacity is demonstrated by the area under the curve (AUC) of 0.92; perfecting categorization is indicated by an AUC of 1.0.the blue curve's obvious ascent over the red-colored dotted line, which represents the initial effectiveness of the randomized classification (AUC = 0.5), indicates that the system perform noticeably better in term of classification accuracy than randomized prediction.

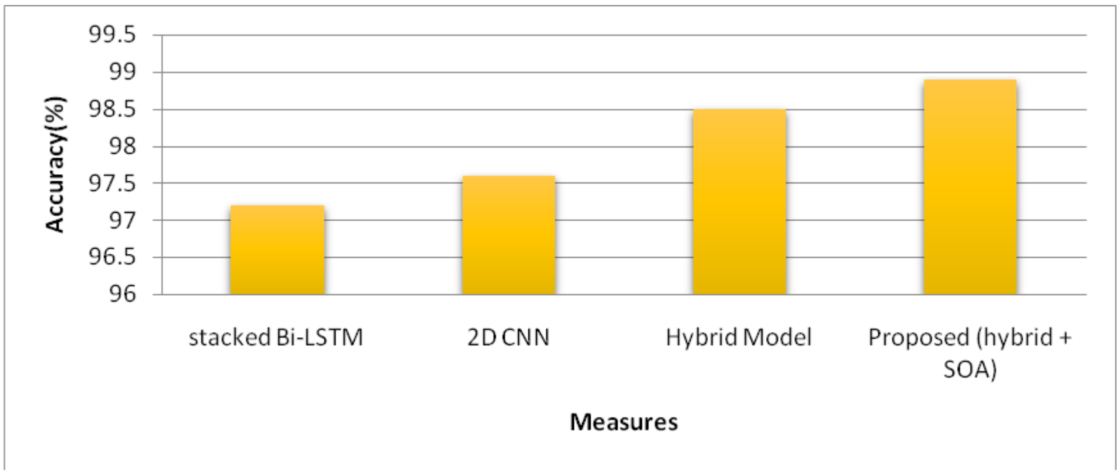

Figure 7: Performance analysis based on Accuracy

Figure 7 presents the accuracy analysis of the proposed approach of fake profile detection where it achieved an impressive accuracy of 98.9% significantly surpassing several existing methods. Compared to stacked Bi-LSTM based classification the proposed technique shows a 2.6% improvement, 2D CNN by 2.4% and it outperforms hybrid model by 0.2%. These results underscore the algorithm's superior effectiveness is accurately finding the fake profiles.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

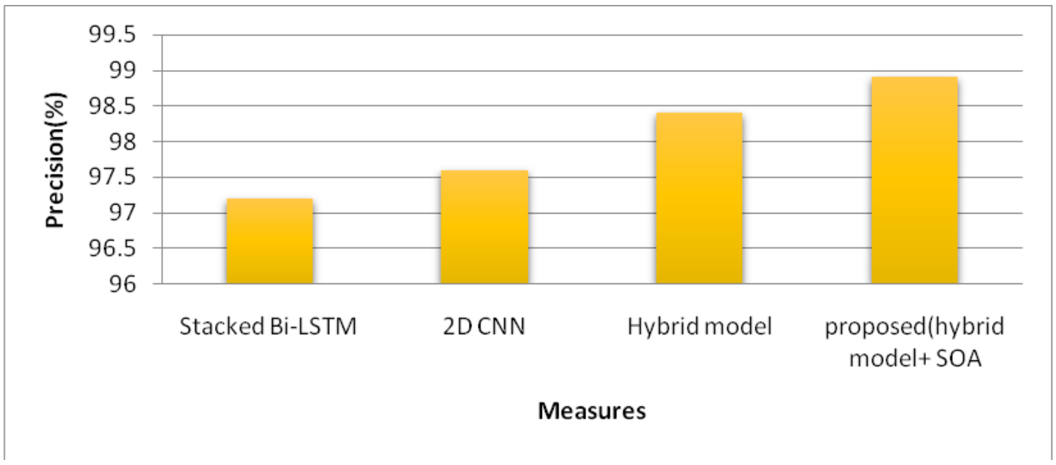

Figure 8: Performance analysis based on Precision

Image 8 presents a precision analysis of the proposed approach in detecting fake profiles, showcasing a remarkable improvement with a precision rate of 0.98. This rate significantly outperforms other techniques, including 0.97 for stacked Bi-LSTM, 0.97 for 2D CNN, and 0.98 for hybrid model. The higher precision underscores the method's effectiveness in accurately identifying true positive cases. Compared to traditional models, the proposed technique offers a more reliable solution for the early detection of fake profiles. This enhanced precision plays a critical role in reducing missed detections, providing a significant advantage over existing approaches.

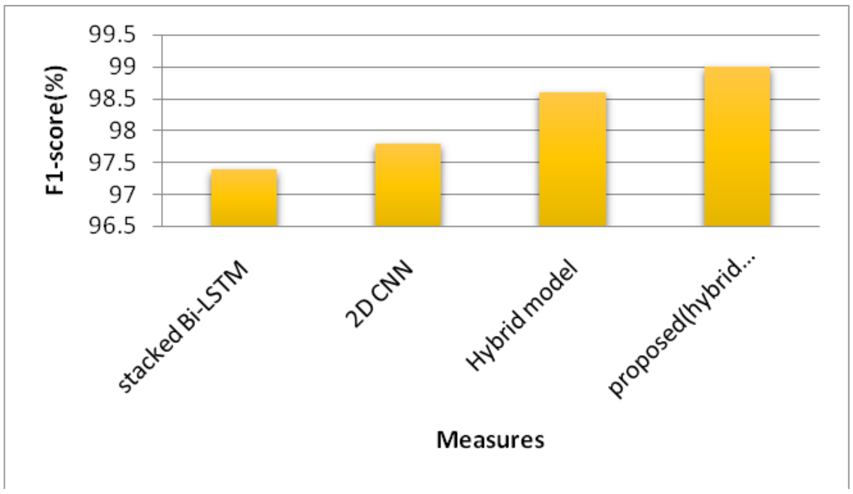

Figure 9: Performance analysis based on F1-score

Figure 9 evaluates the success of the suggested tactics depending on their F1-scores, with the model achieving an impressive 0.99, significantly surpassing existing algorithms. While the lowest F1-score among the compared methods is 0.97 and the highest reaches only 0.98, the proposed model demonstrates exceptional performance in balancing precision and recall. This balance results in higher accuracy for fewer false negatives and more good predictions. The improved F1-score underscores the model's overall reliability and precision, making it a more dependable tool for detecting fake profiles in recommendation systems. By consistently outperforming traditional classification models across key metrics, the approach establishes a new benchmark for diagnostic accuracy.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

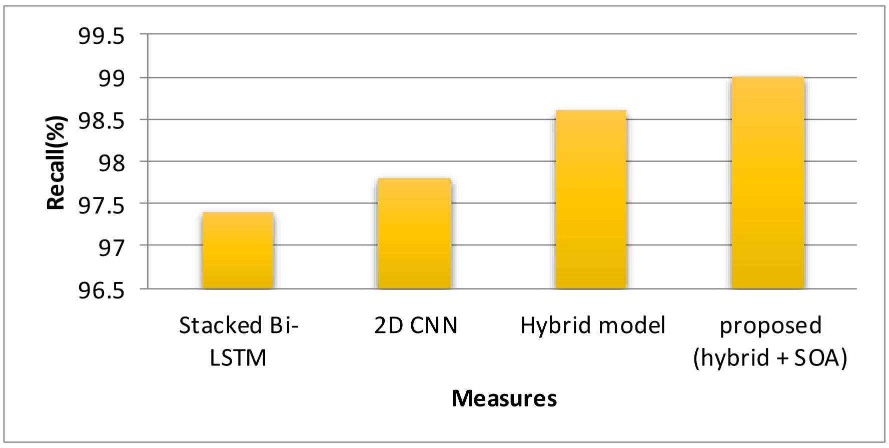

Figure 10: Performance analysis based on Recall

Figure 10 illustrates the classification performance based on recall, highlighting a significant improvement in detecting fake profiles. The proposed method achieves an impressive recall rate of 0.99, outperforming other techniques such as stacked Bi-LSTM with 0.97, 2D CNN with 0.9, and Hybrid model with 0.98. This enhanced recall underscores the method's effectiveness in accurately identifying true positive cases. Compared to traditional models, it demonstrates superior reliability and precision in fake profile detection.

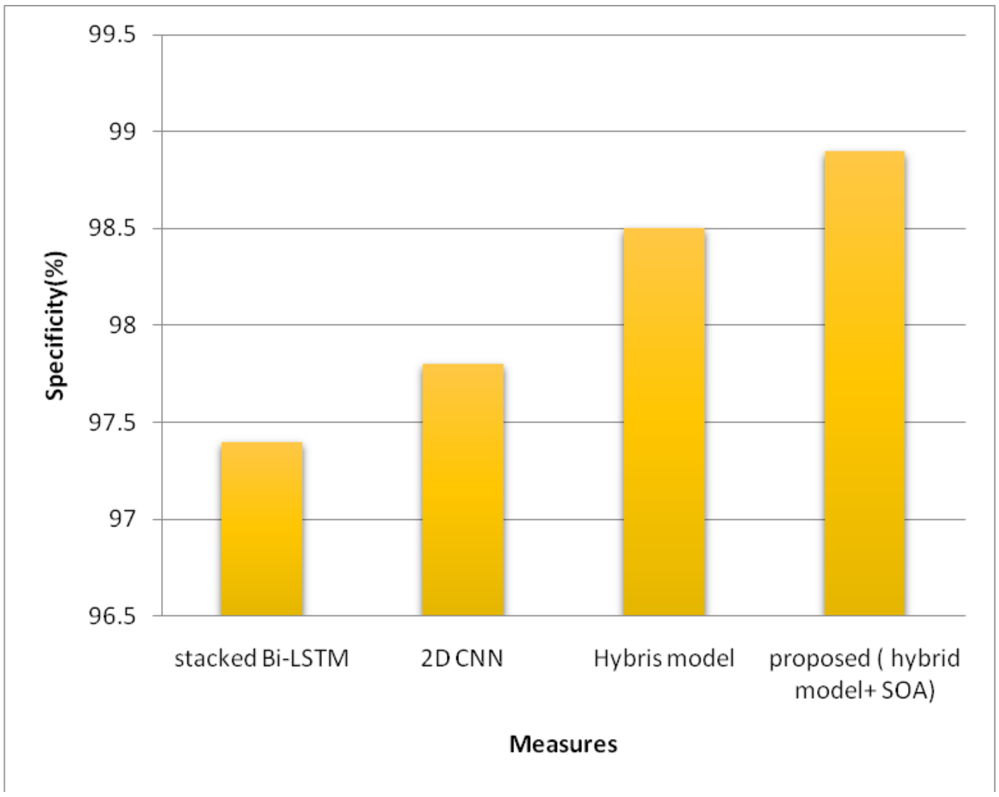

Figure 11: Performance analysis based on Specificity

The suggested approach outperformed other current methods with a specificity of 0.99. In contrast, the hybrid model obtains 0.98 specificity, 2D CNN reaches 0.97, and stacked Bi-LSTM achieves 0.97. This implies the suggested method reduces false positives by more precisely detecting negative cases. As shown in Figure 11 the marked improvement in specificity highlights the method's precision in distinguishing in detecting fake profiles. This advancement provides a distinct advantage over traditional models Provide a better way to detect.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

4.3 Comparison with published work

To show how successful the suggested strategy is, we contrasted our findings using earlier research that has been presented. Specifically, our method was evaluated against studies [17], [19], [21] and [22] which focus on fake profile detection techniques as highlighted in the literature survey. The comparative results, presented in the second table, underscore our method's excellent categorization efficiency achieving an accuracy of 98.9%, our method outperformed the accuracies reported in [17], [19], [21] and [22] which were 97%, 93.5%, 95.5%, and 97.9%, respectively. By employing a hybrid model enhanced with the SOA, our approach consistently demonstrated higher classification accuracy, establishing itself as a robust and reliable solution for precise data classification.

Table 2: Comparative analysis results

Reference	Technique	Accuracy %	Specificity %	Precision %	F1-Score %	Re-call %
Ajesh, F. et al.[17]	NLP	97%	Not measured	95.7%	93.7%	80%
Sahoo, S. R., and Gupta, B. B. et al.[19]	machine learning-based approach	93.5%	Not measured	98.4%	Not measured	96.7%
Wanda, P., and Jie, H. J et al.[21]	DNN	95.5%	98.1%	97.9%	Not measured	Not measured
Masood, F. et al.[22]	URL	97.9%	Not measured	Not measured	96.8%	98.1%
Proposed	Hybrid model+SOA	98.9%	98.9%	98.9%	99%	995

CONCLUSION

5. CONCLUSION

The study focused on detecting fake profiles through a framework comprising preprocessing, feature extraction, and classification stages. During preprocessing, duplicate entries were removed, missing values were handled using mean imputation, and data consistency was ensured through Min-Max normalization. For feature extraction, Principal Component Analysis (PCA) was employed to reduce data dimensionality. The classification framework combined a 2D Convolutional Neural Network (2D CNN) to capture spatial features and structural patterns, with an Attention-Enhanced Stacked BiLSTM (AES-BiLSTM) for sequential feature analysis enhanced by attention mechanisms. To further improve performance, the hyperparameters of both the 2D CNN and AES-BiLSTM models were optimized using the Seagull Optimization Algorithm (SOA). Implemented in Python, this approach achieves high accuracy (98.9%), precision (98.9%), specificity (98.9%), and an F1-score of 99%, offering a dependable and expandable method for efficiently identifying phony social networking accounts. In order to detect fraudulent information across diverse profiles, future research will employ a variety of neuronal network-based feature analysis methods. Implementing a limitation method through the creation of aggregators modules which analyzes profiles behaviors and attributes would make an efficient way to protect the network against fraudulent accounts.

REFRENCES

[1] Luttrell, R. (2018). Social media: How to engage, share, and connect. Rowman & Littlefield.
[2] Schreck, T., & Keim, D. (2012). Visual analysis of social media data. Computer, 46(5), 68-75.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

[3] Jethava, G., & Rao, U. P. (2024). Exploring security and trust mechanisms in online social networks: An extensive review. Computers & Security, 103790.
[4] jalal yousef Zaidieh, A. (2024). Combatting Cybersecurity Threats on Social Media: Network Protection and Data Integrity Strategies. Journal of Artificial Intelligence and Computational Technology, 1(1).
[5] Kayes, I., & Iamnitchi, A. (2017). Privacy and security in online social networks: A survey. Online Social Networks and Media, 3, 1-21.
[6] Rao, S., Verma, A. K., & Bhatia, T. (2021). A review on social spam detection: Challenges, open issues, and future directions. Expert Systems with Applications, 186, 115742.
[7] Kumar, S., & Shah, N. (2018). False information on web and social media: A survey. arXiv preprint arXiv:1804.08559.
[8] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), 22-36.
[9] Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673.
[10] Rahmani, S., Aghalar, H., Jebreili, S., & Goli, A. Optimization and computing using intelligent data-driven approaches for decision-making. In Optimization and Computing using Intelligent Data-Driven Approaches for Decision-Making (pp. 90-176). CRC Press.
[11] Alzaabi, F. R., & Mehmood, A. (2024). A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. IEEE Access, 12, 30907-30927.
[12] Jebreili, S., & Goli, A. (2024). Optimization and computing using intelligent data-driven. Optimization and Computing using Intelligent Data-Driven Approaches for Decision-Making: Optimization Applications, 90.
[13] Sarker, I. H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. Annals of Data Science, 10(6), 1473-1498.
[14] Ofoegbu, K. D. O., Osundare, O. S., Ike, C. S., Fakeyede, O. G., & Ige, A. B. (2024). Real-Time Cybersecurity threat detection using machine learning and big data analytics: A comprehensive approach.
[15] Tufchi, S., Yadav, A., & Ahmed, T. (2023). A comprehensive survey of multimodal fake news detection techniques: advances, challenges, and opportunities. International Journal of Multimedia Information Retrieval, 12(2), 28.
[16] Kadam, N., & Sharma, S. K. (2022). Social media fake profile detection using data mining technique. Journal of Advances in Information Technology Vol, 13(5).
[17] Ajesh, F., Aswathy, S. U., Philip, F. M., & Jeyakrishnan, V. (2021). A hybrid method for fake profile detection in social networkusing artificial intelligence. Security Issues and Privacy Concerns in Industry 4.0 Applications, 89-112.
[18] Amankeldin, D., Kurmangaziyeva, L., Mailybayeva, A., Glazyrina, N., Zhumadillayeva, A., & Karasheva, N. (2023). Deep Neural Network for Detecting Fake Profiles in Social Networks. Computer Systems Science & Engineering, 47(1).
[19] Sahoo, S. R., & Gupta, B. B. (2020). Fake profile detection in multimedia big data on online social networks. International Journal of Information and Computer Security, 12(2-3), 303-331.
[20] Aditya, B. L., & Mohanty, S. N. (2023). Heterogenous Social Media Analysis For Efficient Deep Learning Fake-Profile Identification. IEEE Access.
[21] Wanda, P., & Jie, H. J. (2020). DeepProfile: Finding fake profile in online social network using dynamic CNN. Journal of Information Security and Applications, 52, 102465.
[22] Masood, F., Almogren, A., Abbas, A., Khattak, H. A., Din, I. U., Guizani, M., & Zuair, M. (2019). Spammer detection and fake user identification on social networks. IEEE Access, 7, 68140-68152.
[23] Murugan, S., Daniel, K. L., Bernice, A. T., Gunasekaran, V., & Anbumani, A. (2024). Fake Social Media Profile Detection Using Ml Algorithms. V. and Anbumani, A., Fake Social Media Profile Detection Using Ml Algorithms (December 19, 2024).

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.