Optimizing Bidirectional LSTM Networks with Temperature-Scaled Sigmoid Activation for Enhanced Fake Profile Detection on Social Media

Govind Singh Mahara
RKDF University
Bhopal
India

Sharad Gangele
RKDF University
Bhopal
India

Mangala Prasad Mishra
Indira Gandhi National Open University
New Delhi

ABSTRACT

This study presents an optimized Bidirectional Long Short-Term Memory (Bi-LSTM) network that integrates a temperature-scaled softmax function to improve the detection of fake profiles on Instagram and other social media platforms. By incorporating temperature scaling, the model effectively reduces overconfidence in its probability predictions, leading to more calibrated and reliable outputs. [1] Is explained by temperature scaling in single parameter variation, In this setup, the temperature-scaling function acts as a modifier on the softmax output layer, which enhances the model’s confidence alignment and mitigates issues that arise from predictions that might otherwise be excessively certain, particularly in ambiguous cases.The research explores the impact of varying learning rates and temperature values across multiple experimental setups. By fine-tuning these parameters, the model achieves an optimized balance in accuracy and stability, enhancing its overall performance. The Bi-LSTM model outperformed traditional methods in several key metrics, including accuracy, precision, recall, and F1 score. This suggests that the temperature-scaled Bi-LSTM approach is better suited for tasks that require nuanced classification in binary settings, such as differentiating between real and fake profiles. Additionally, the proposed framework’s flexibility allows it to be readily adapted to other binary classification problems on social media platforms, including spam filtering and fraud detection. This adaptability extends the potential applications of the model across various domains where binary classification is essential for safeguarding platform integrity.The experiments for this research were conducted on Google Colab, a popular cloud-based platform offering free access to GPUs, making it a suitable environment for deep learning projects

General Terms

Pattern, Recognition, Security, Algorithms, Machine Learning, Deep Learning, Binary Classification, Fraud Detection, Spam, Filtering, Artificial Intelligence, Natural Language Processing (NLP).

Keywords

Fake User, Deep learning, RNN, LSTM, Bi-LSTM, Neural Network, Softmax, learning rate, Temperature-scaling, Classification, Social media, Instagram profiles .

1. INTRODUCTION

Fake users, commonly known as fake accounts, are social media profiles created and managed by either automated bots or real individuals with deceptive intentions. These accounts are often designed to manipulate, deceive, or disrupt online conversations. Fake users can engage in spamming, phishing, spreading propaganda, and manipulating public opinion, which distorts online dialogue and contributes to the misinformation ecosystem [2].They can also be used to harass genuine users, misappropriate personal information, or serve specific agenda-driven objectives that undermine the trust and value of social media platforms[3]. The proliferation of fake accounts poses significant challenges, affecting not only the user experience but also the credibility and functionality of these platforms[4].Detecting and removing fake users has become increasingly crucial to maintaining a trustworthy online environment. Research has found that fake accounts are highly prevalent, constituting an estimated 9-15% of active social media profiles, with platforms such as Twitter and Facebook reporting the highest percentages of fake users [5]. These accounts contribute significantly to overall activity on social media, producing nearly 48% of tweets and approximately 5% of posts on Facebook [4] Recently, researchers found that many fake accounts are even adopting AI-generated profile pictures to appear more authentic, further complicating the detection process [6]. This tactic not only misleads real users but also makes it harder for detection algorithms to differentiate between real and fake profiles.

1.1 Key approaches to Fake User Detection

The task of identifying fake users involves several approaches, each addressing different aspects of the problem with unique strengths and limitations. The primary methods include machine learning algorithms, manual verification, and behavior analysis. Each approach leverages a distinct combination of technical and behavioral insights to detect fake profiles more accurately.

1.1.1 Machine Learning Algorithms

Machine learning models are instrumental in fake user detection, leveraging extensive datasets of both real and fake profiles to classify accounts. Popular methods include traditional classifiers like decision trees and support vector machines, as well as advanced neural networks such as LSTM and CNN models [7]. These machine learning systems analyze vast amounts of data to learn patterns distinguishing fake users from genuine ones, often achieving high accuracy in controlled datasets. However, the effectiveness of these models depends significantly on the quality and representativeness of the training data. Models trained on biased or incomplete datasets may struggle to generalize to new or unseen profiles, limiting their scalability and reliability in real-world scenarios [3].

1.1.2 Manual Verification

Human analysis remains a critical, though resource-intensive, method for identifying fake accounts. Human reviewers assess accounts based on established criteria, effectively spotting suspicious accounts through nuanced judgment that often eludes automated methods [5]. While manual verification can achieve high accuracy, it is time-consuming and impractical for large-scale applications, making it more suitable as a supplementary measure to other automated techniques.

1.1.3 Behavior Analysis

This approach examines user activity patterns, such as the ratio of followers to followed accounts, posting frequency, and engagement in discussions. Behavior analysis can identify accounts with traits often associated with fake users, such as disproportionately high follower counts, excessive posting of promotional or suspicious content, and inconsistent activity patterns [8] Although behavior analysis offers valuable insights, it can be limited by increasingly sophisticated fake accounts capable of closely mimicking genuine user behavior, making detection more challenging [7].

Given these limitations, researchers have begun exploring advanced models, such as Bidirectional LSTM networks combined with temperature-scaled softmax activation functions, to refine the accuracy of fake account detection. Temperature scaling helps mitigate overconfidence in predictions by adjusting the softmax output, a common issue in traditional neural networks [4]. Testing different learning rates and temperature settings further enables researchers to optimize model performance across various datasets and platforms.

1.2 Characteristics and Influence of Fake Users

Fake accounts often display identifiable characteristics, such as large numbers of followers, excessive spamming activities, minimal personal information, and erratic posting patterns [3]. These profiles may also exhibit bursts of activity, such as multiple posts in a short timeframe or sustained high-volume posting, to amplify specific content quickly. However, detecting fake users based on these traits alone is challenging because sophisticated fake accounts increasingly adopt behaviors that mimic real users [3].

The influence of fake users extends beyond simple disruptions in social media; they shape online conversations and public opinion. Some fake accounts, for example, serve commercial interests by spamming and advertising, while others spread misinformation or propaganda to influence political or social narratives [4]. Additionally, fake users manipulate public perception by artificially inflating likes, comments, or shares on particular posts, creating the illusion of popularity or credibility. Such tactics not only distort online dialogue but also undermine trust in social media as a reliable source of information, which can impact both individual users and society as a whole [7].

2. LITERATURE REVIEW

There has been a significant amount of research published on the topic of fake user detection in social media. [9] This paper explore post-hoc calibration techniques for neural networks, particularly focusing on confidence scaling through an Adaptive Temperature Scaling approach, authors introduce Entropy-based Temperature Scaling, a method that adjusts prediction confidence based on the entropy of predictions, linking entropy to overconfidence levels.[10] introduce a method to improve temperature scaling in autoregressive models, which are traditionally limited by myopic (short-sighted) temperature adjustments that optimize only the next token prediction. Temperature scaling is widely used to control model sharpness, calibrate uncertainty, and adjust sampling probabilities, especially in large language models [11] This paper presents the foundational use of Bidirectional LSTMs (Bi-LSTMs) for sequence classification and demonstrates their effectiveness in tasks that require both forward and backward context, This research paper uses the concept of using Bidirectional Long Short-Term Memory (Bi-LSTM) networks for frame wise phoneme classification. The study explored different neural network architectures, showcasing the advantage of Bi-LSTMs in speech recognition tasks due to their ability to utilize both past and future context information, leading to superior performance over unidirectional models. The findings have laid foundational groundwork for sequence-based classification tasks in machine learning and deep learning applications. [12] This work introduces the original concept of bidirectional RNNs, laying the groundwork for their subsequent adaptation into Bi-LSTMs and Bi-GRUs, This seminal work introduced Bidirectional Recurrent Neural Networks (Bi-RNNs), establishing a framework for neural networks to process data in both forward and backward directions. By maintaining two separate hidden states for each direction, Bi-RNNs significantly enhanced the capability to capture temporal dependencies, making them particularly effective in speech and text processing tasks. This approach paved the way for more sophisticated architectures like Bi-LSTM and Bi-GRU. [13] The paper discusses the implementation of deep learning models, including LSTMs and their variations, for time series classification, which is relevant to sequence tagging and pattern recognition, The authors proposed a method for time series classification using deep neural networks trained from scratch. Their work provided a strong baseline by demonstrating that deep learning models can outperform traditional machine learning techniques on time series data without requiring extensive feature engineering. This approach has set a benchmark for future time series classification research. [14] This research integrates Bi-LSTMs with character-level features to improve language modeling tasks, which can be adapted for other applications like social media profile detection., The paper presented a novel character-aware neural language model, which captures word-level and character-level features to improve language modeling tasks. By incorporating character-level features, the model enhances its ability to understand morphology and spelling variations, resulting in improved performance on downstream tasks like language modeling and text classification [15] This paper explores multi-scale feature extraction, which can be combined with Bi-LSTMs to enhance their performance in hierarchical time-series analysis, The authors introduced Multi-Scale Convolutional Neural Networks (MC-CNNs) for time series classification. This method effectively captures temporal dependencies at multiple scales, thereby enhancing the model's robustness and performance. The proposed architecture demonstrates superior results over traditional CNNs and RNNs in handling time series data. [16] The combination of LSTMs and convolutional networks in this study offers insights into building hybrid models, which can be beneficial for improving Bi-LSTM-based architectures. This study proposed LSTM Fully Convolutional Networks (LSTM-FCNs) for time series classification, combining the strengths of LSTMs and CNNs. LSTM layers capture long-term dependencies, while FCNs extract high-level temporal features, leading to improved performance on diverse time series classification tasks. [17]

This paper provides a comprehensive review of RNNs and their variants, discussing their limitations and strengths in sequence learning tasks. [18] This seminal work introduced the LSTM architecture, which solved the vanishing gradient problem in RNNs and paved the way for advanced LSTM-based models, The authors introduced the Long Short-Term Memory (LSTM) architecture. [19] This study proposes a multi-channel approach combining LSTM and FCN for improved time-series classification. The paper proposed a Multi-Channel LSTM-FCN architecture using the MACD-histogram for time series classification [20]. This paper explored the use of FCN-BiLSTM architecture for VAT invoice recognition. The combination of Fully Convolutional Networks (FCN) and BiLSTM allows for efficient processing of sequential invoice data, demonstrating the model's effectiveness in automating complex document recognition tasks.[21] The study proposed a Bi-LSTM approach for fake news detection, leveraging the model's ability to capture context from both directions in a sequence.. [22] It provides an in-depth analysis of existing challenges, such as data imbalance and misinformation spread, and proposes potential directions for improvement. The study emphasizes the role of advanced AI techniques in enhancing the accuracy and reliability of fake news detection systems [23] This paper explores the application of recurrent neural networks (RNN), long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM) models for the task of fake news detection. The authors present a comprehensive analysis of how these deep learning architectures can be leveraged to identify and classify fake news content, highlighting the effectiveness of Bi-LSTM networks in capturing temporal dependencies and contextual information from text data. The results indicate that Bi-LSTM models outperform traditional methods, showcasing their robustness in handling complex sequences and enhancing the accuracy of fake news detection[24] This paper investigates the impact of user profiles on the effectiveness of fake news detection algorithms. The authors analyze how user characteristics, such as activity patterns, social interactions, and historical behavior, can be utilized to improve the performance of detection models. By incorporating features derived from user profiles, the study demonstrates enhanced capability in distinguishing between genuine and fake news. The research further highlights the importance of understanding the user context in order to design more robust and accurate detection systems. This work provides valuable insights into user-centric approaches, suggesting that integrating user profile information can significantly boost the detection of fake news on social media platforms.There are a few gaps or limitations in the current literature on fake user detection in social media that suggest areas for future research:

Lack of robust evaluation methods: Many of the existing studies on fake user detection have used small or synthetic datasets, which may not be representative of the overall population of accounts. Additionally, there is often a lack of standardization in the evaluation methods used, which makes it difficult to compare the results of different studies.

1. Limited focus on specific platforms
2. Limited consideration of user motivations
3. Lack of attention to legal and ethical issues

Overall, there is a need for more research on fake user detection in social media that is based on larger and more diverse datasets, that is more generalizable across different platforms, that takes into account the social aspects of fake accounts, and that addresses the legal and ethical implications of fake user detection. The proposed research, the key questions being addressed are:

1. How can the effectiveness of Bi-LSTM networks be improved for detecting fake profiles on social platforms?

2. How does varying learning rates influence the performance of the Bi-LSTM model?

3. What is the comparative impact of temperature scaling on different performance metrics?

4. Can temperature-scaled softmax provide more reliable probability estimates for fake profile detection?

5. How does the proposed method compare to traditional fake profile detection techniques?

3. PROPOSED MODEL

The effectiveness of machine learning and deep learning approaches for detecting fake user profiles on social media platforms is highly dependent on several factors, including the quality of training data, the complexity of the model, and the specific context in which the algorithm operates. A robust model should be capable of distinguishing genuine users from fraudulent ones while maintaining a high level of generalizability across different datasets and platforms.

In this study, an optimized Bidirectional Long Short-Term Memory (Bi-LSTM) network is proposed as a solution for identifying fake profiles with enhanced accuracy and reliability. The proposed approach integrates a temperature-scaled softmax function, which plays a crucial role in probability calibration—an essential factor in ensuring the reliability of machine learning-based classification models. Temperature scaling helps reduce the problem of overconfident predictions, a common issue in deep learning models, thereby improving the model's ability to differentiate between real and fake accounts with greater certainty.

3.1 Methodology and Experimental Setup

The methodology in this paper focuses on analyzing the impact of different learning rates and temperature values on the model's overall performance. A structured approach is taken to evaluate various configurations of these hyperparameters. Specifically, the learning rates considered are 0.003, 0.005, and 0.009, while temperature values are varied across 0.1, 0.3, 0.5, and 0.9. This results in a total of 12 distinct experimental configurations, each providing valuable insights into the effects of hyperparameter tuning on model accuracy, precision, recall, and F1-score.

3.2 Role of Bi-LSTM in Fake Profile Detection

The choice of a Bi-LSTM architecture is motivated by its ability to capture temporal dependencies in user activity patterns, leveraging both forward and backward sequences to extract comprehensive contextual information. Unlike traditional models that process user activity data sequentially in a single direction, Bi-LSTM processes information from both past and future contexts, making it particularly effective in identifying complex behavior patterns associated with fake profiles.

3.3 Temperature Scaling and Probability Calibration

One of the key challenges in deep learning-based classification is overconfident predictions, where models assign extremely high probabilities to their predictions, even in cases of ambiguity. This can lead to poor decision-making in real-world applications. By integrating temperature scaling within the softmax function, this study aims to calibrate the output probabilities, ensuring that they align better with real-world distributions. This results in more reliable confidence scores, reducing the risk of misclassifying genuine users as fake or vice versa.

3.4 Experimental Findings and Model Performance

The experimental results demonstrate that the temperature-scaled Bi-LSTM model consistently outperforms traditional softmax-based models across different configurations. The evaluation metrics—accuracy, precision, recall, and F1-score—indicate robust model performance, confirming the effectiveness of the proposed approach. Among the tested hyperparameter combinations, the learning rate of 0.009 combined with a temperature of 0.9 yielded the highest classification accuracy, further validating the importance of fine-tuning hyperparameters to optimize model performance.

3.5 Modeling Fake User Profile Detection Using Bi-LSTM with Sigmoid Function

The dataset used for training and testing the model was sourced from Kaggle, leveraging its high-quality, structured data to create a realistic test bed for evaluating the model's performance. The given dataset https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts/data] consists of several attributes that describe user profiles, such as profile pic, nums/length username, fullname words, and so on. Flowing attributes are included in the feature selection (profile pic, numbs/length username, fullname words, nums/length fullname, name==username , description length , external URL, private, posts, followers, follows, fake), The goal is to model these attributes as a sequence to detect whether a profile is fake or genuine using a Bidirectional Long Short-Term Memory (Bi-LSTM) network. The Bi-LSTM network is chosen because it effectively captures sequential dependencies in the data, leveraging both forward and backward contexts. Additionally, In this paper apply the Sigmoid function with varying temperatures to control the confidence of the predictions and mitigate overconfidence.

3.5.1 Steps for Modeling

(a) Define Sequential Features: Each attribute in the dataset is treated as a sequential feature, similar to words in a sentence. The features include:

1. profile pic: Binary attribute indicating if a profile picture is present.
2. nums/length username: Ratio of numeric characters in the username.
3. fullname words: Number of words in the user's full name.
4. nums/length fullname: Ratio of numeric characters in the full name.
5. name==username: Binary value indicating whether the full name matches the username.
6. description length: Length of the profile description.
7. external URL: Binary value indicating if an external URL is present.
8. private: Binary value indicating if the profile is private.
9. #posts: Number of posts made by the user.
10. #followers: Number of followers.
11. #follows: Number of profiles the user follows.
12. fake: Target label indicating whether the profile is fake (1) or genuine (0).

(b) Prepare the Data as Sequential Input: Each profile is represented as a sequence of attributes $x_{1,}x_{2,}\dots,x_n$ with the label y as the target output.

1. **Example Sequence**:
$$x = [1,0.27,0,0.00,0,53,0,0,32,1000,955]$$

Here, x is a feature vector representing a single profile.

(c) Mathematical Representation:

For each profile, To predict the probability of it being fake:

$p(y = fake|x_1, x_2, \dots, x_{11})$ ….……………………… eq (1)

Using Bayes' theorem, Can express this posterior probability as:

$p(y = fake|x_1, x_2, \dots, x_{11}) = \frac{p(x_1,x_2,\dots,x_{11}\,|y=fake|).p(y=fake)}{p(x_1,x_2,\dots,x_{11})}$ …………eq (2)

Where:

$p(x_1, x_2, \dots, x_{11} \mid y = fake\ )$ The likelihood of observing the attributes given the label y.

$p(y)$ Is the prior probability of a profile being fake or genuine.

(d) Sigmoid Function:

The sigmoid function is used in the output layer of the Bi-LSTM model to convert logits (raw outputs) into probabilities[.The sigmoid function is defined as [43]:

$\sigma(x) = \frac{1}{1+e^{-x}}$ …………………….. eq (3)

Where:

x represents the input logits.

The sigmoid function maps any input x to a value between 0 and 1, which can be interpreted as a probability. For example, if $\sigma(x) = 0.7$, then there is a 70% probability that the profile is fake.

(e) Temperature Scaling: To control the sharpness of the sigmoid output, temperature scaling can be applied. The temperature-scaled sigmoid is given by:

$\sigma(x, T) = \frac{1}{1+e^{-\frac{x}{T}}}$ ……………………… eq (4)

Where:

T is the temperature parameter.

Lower T values produce sharper probability distributions, making the model more confident in its predictions.

Higher T values produce smoother probability distributions, reducing overconfidence.

By adjusting the temperature T, It can control how confidently the model makes its predictions.

(f) Utilizing Bi-LSTM for Sequence Learning: The Bi-LSTM model processes the sequential attributes in both forward and backward directions, learning dependencies and interactions among attributes. The model architecture is defined as:

1. **Input Layer**: Takes the sequence of attributes as input.
2. **Bi-LSTM Layer**: Processes the sequence in both forward and backward directions to capture all dependencies.
3. **Fully Connected Layer**: Outputs a single value representing the probability of the profile being fake.
4. **Temperature-Scaled Sigmoid Function**: Applies temperature scaling to the output of the fully connected layer to control the confidence level of the probability predictions.

(g) Sequence and Probability Modeling:

1. **Forward Sequence**:

   $p(x_1, x_2, \ldots, x_{11} \mid y)$……… eq (5)

   This represents the probability of observing the sequence x_{11} to x_1 given y (fake or genuine) in the forward direction.

2. **Backward Sequence**:

   $p(x_{11}, x_{10}, \ldots, x_1 \mid y)$……….eq (6)

   This represents the probability of observing the sequence to x1x_1x1 given yyy (fake or genuine) in the backward direction.

3. **Joint Probability**:

$$p(x|y) = p(x_1, x_2, \ldots, x_{11} \mid y).\, p(x_{11}, x_{10}, \ldots, x_1 \mid y)$$

….. eq (7)

The joint probability combines the forward and backward sequences.

(h)Bi-LSTM Model Implementation:

A Bidirectional Long Short-Term Memory (Bi-LSTM) network is a type of recurrent neural network (RNN) designed to effectively capture contextual information in sequences. Unlike a standard LSTM, which processes information only in a forward direction, a Bi-LSTM network processes the input sequence in both forward and backward directions. This dual-layer structure allows the network to leverage information from both past and future contexts, making it particularly powerful for understanding the relationships between words and phrases in both directions within a sequence.The Bi-LSTM architecture consists of two LSTM layers. One LSTM layer processes the sequence from start to end (forward direction), while the other processes it from end to start (backward direction). The outputs from these two layers are then combined using operations like averaging, summation, multiplication, or concatenation to form a unified representation of the sequence. This bidirectional structure enables the model to have a more comprehensive understanding of the sequential dependencies, making it highly effective for tasks such as natural language processing, time-series analysis, and anomaly detection.

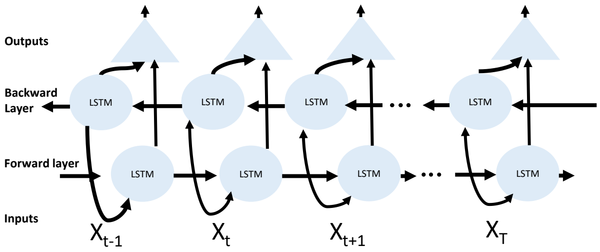

Fig:1 Bidirectional Long Short-Term Memory (Bi-LSTM) network architecture

In an unrolled Bi-LSTM, the network's architecture shows two parallel LSTM networks—one for each direction—followed by a merging layer that consolidates information from both directions to produce the final output.

3.6 Model creation with Sigmoid Activation Function

This section of the research paper explores the implementation of a Bidirectional Long Short-Term Memory (Bi-LSTM) model using Python for classifying Instagram users as fake or genuine. The classification is based on a diverse set of features extracted from user profiles, including numerical, textual, and behavioral attributes. These features encompass metrics such as followers-to-following ratios, posting frequency, engagement levels, and text-based characteristics from user bios and captions.

The Bi-LSTM architecture is chosen due to its ability to process sequential data bidirectionally, leveraging both forward and backward dependencies to identify intricate patterns in user behavior. Unlike standard LSTMs, which analyze only past data points, Bi-LSTMs enhance feature representation by incorporating future context, allowing for a more nuanced classification of social media profiles.

By effectively capturing contextual relationships between features, the Bi-LSTM model significantly improves its ability to differentiate between real and fake profiles. This architecture proves particularly beneficial in scenarios where fraudulent accounts mimic authentic user behavior, as the bidirectional processing ensures that even subtle discrepancies are detected. The proposed model is implemented using Python and deep learning libraries such as TensorFlow and Keras, optimizing classification accuracy through hyperparameter tuning and probability calibration techniques.

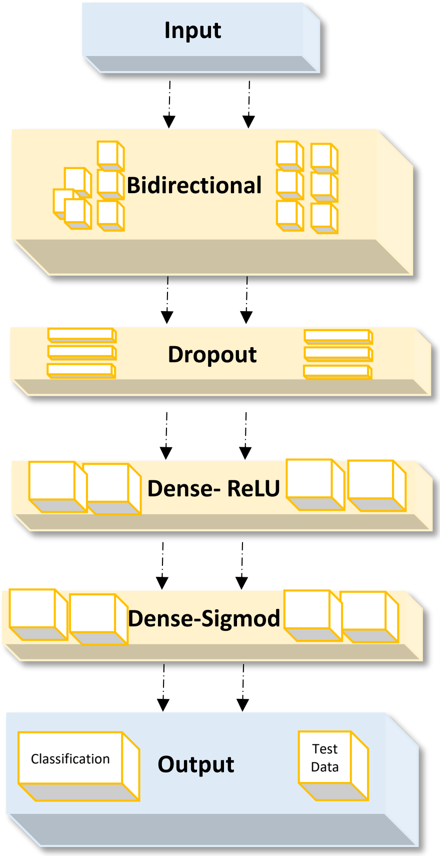

Fig:2 Bi-LSTM Model.

To perform implementation and execution google colab is used, with high ram and TPU.

BI-The model layers can be described as follows.

1. **Bidirectional LSTM Layer:** Since the LSTM is bidirectional, it processes sequences both forward and backward, and the hidden states from both directions are concatenated, [25] the parameters represents the total number of learnable parameters in the LSTM layer. The parameters include weights for the input, recurrent weights, and biases.

In a bidirectional LSTM, there are two sets of LSTMs (one for the forward pass and one for the backward pass), which explains the large number of parameters.

Following formula for LSTM parameters.

$$LSTM\ Parameters = 4 \times [(input\ size \times hidden\ size) + (hidden\ size \times hidden\ size) + hidden\ size]$$

Where 4 represents the four gates (input, forget, output, and candidate).

2. **Dropout Layer:** This layer doesn't change the shape of the data. It simply drops a percentage of the neurons (set by a dropout rate, typically between 0.2 and 0.5) during training to prevent over fitting.

3. **Dense Layer:** This layer reduces the dimensionality of the input from 2000 to 64. The fully connected (dense) layer applies linear transformations to the data.

   Formulation of Dense Layer Parameters

   $$Dense\ Parameters = (input\ size \times output\ size) + output\ size$$

4. **Dense Layer:** This is the final output layer, and since this is a binary classification task (such as fake profile detection), the output is a single neuron with a value between 0 and 1, representing the probability of being fake.

This architecture effectively captures sequential dependencies and outputs the probability of whether a given profile is fake or not.

3.6.1 Flow for prediction with Temperature-Scaled Sigmoid Function

The application of temperature scaling to the sigmoid function in binary classification is discussed in the[1]This discussion provides insights into adapting temperature scaling for models that utilize the sigmoid function in binary classification tasks.

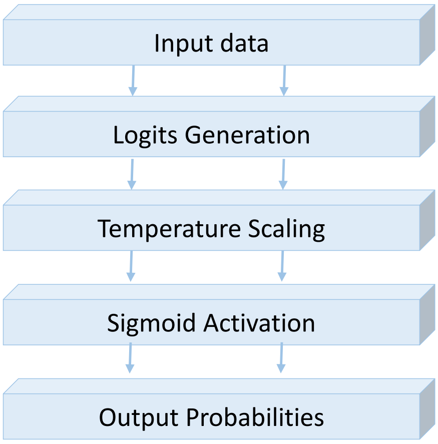

Fig:3 Temperature scaling with Sigmoid Activation.

1. **Input Data**: The model takes in input data (features from the dataset).
2. **Logits Generation**: The model generates logits (raw prediction scores).
3. **Temperature Scaling**: The logits are divided by the temperature value (T).
4. **Sigmoid Activation**: The temperature-scaled logits are passed through the sigmoid function to produce probabilities.
5. **Output Probabilities**: The final output is the set of probabilities that represent the prediction.

4. PERFORMACE EVALUATION

The model's performance is evaluated using the following metrics:

4.1 ROC Curve and AUC: The Receiver Operating Characteristic (ROC) Curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. These metrics are defined as:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Where:

- **TP (True Positives)**: Number of correctly predicted positive instances.
- **FP (False Positives)**: Number of incorrectly predicted positive instances.
- **FN (False Negatives)**: Number of incorrectly predicted negative instances.
- **TN (True Negatives)**: Number of correctly predicted negative instances.

The Area Under the ROC Curve (AUC) is a single scalar value that summarizes the performance of the model over all possible threshold values. It is computed as:

$AUC = \int_0^1 \mathrm{TPRd(FPR)}$ ………………… eq (1)

The AUC value ranges from 0 to 1, where a value closer to 1 indicates better classification performance.

4.2 Confusion Matrix: A confusion matrix provides a summary of the model's classification performance by displaying the counts of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). It is structured as follows:

$$\text{Confusion Matrix} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

Each element represents the count of model predictions as compared to the actual labels:

TN: True Negatives - Correctly predicted negatives.

FP: False Positives - Incorrectly predicted as positive.

FN: False Negatives - Incorrectly predicted as negative.

TP: True Positives - Correctly predicted positives.

4.3 Precision-Recall Curve: The **Precision-Recall Curve** is used when the data is imbalanced and focuses on the trade-off between precision and recall. These metrics are defined as[26]:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

[27]The precision-recall curve is plotted with recall on the x-axis and precision on the y-axis at different threshold values. The Area Under the Precision-Recall Curve (PR-AUC) is defined as:

$\mathrm{PR} - \mathrm{AUC} = \int_0^1 \text{Precision } d(\text{Recall})$ ……………..eq(1)

The higher the PR-AUC value, the better the model performs, particularly in identifying the minority class.

4.4 Training Loss Progression: The training loss monitors the performance of the model during training by evaluating the difference between the predicted output and the actual labels. The loss function, such as Binary Cross-Entropy (for binary classification),[26] is defined as:

$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i.\log(\hat{y}_i) + (1 - y_i).\log(1 - y_i)]$--- eq (1)

Where:

- N is the number of samples.
- y_i is the true label (0 or 1).
- $\hat{y}_i$ is the predicted probability for label 1.

The loss is calculated for each epoch during training and plotted as a graph, helping to visualize how well the model is learning over time.

5. PERFORMACE EVALUATION

Data preprocessing is a critical step to ensure that the model receives data in an appropriate format. The dataset used in this research contains features related to social media account activities and attributes.

- **Normalization and Standardization:** Continuous features are standardized using

Fig 4(a) The figure aims to provide an overview of the distribution and patterns within the dataset, focusing on the distinction between private profiles and fake profiles

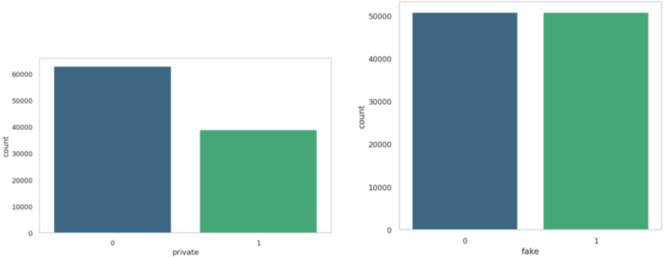

Fig:4 (a)Data visualization data set (Private and Fake Profile)

Fig-5 (b) Profiles identified as fake based on labels in the dataset. Their distribution highlights behavioral patterns or feature values distinct from genuine profiles.

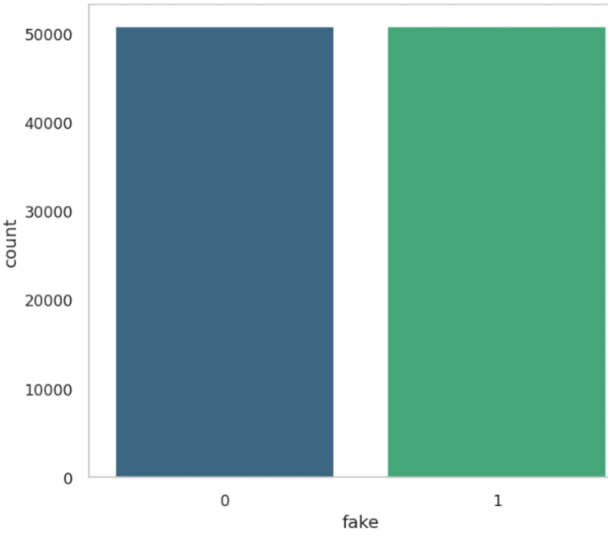

Fig:5(b) Data visualization data set (Fake profile)

Fig 5(c) Profiles marked as private in the dataset. Their distribution is shown to compare with the patterns observed in fake profiles.

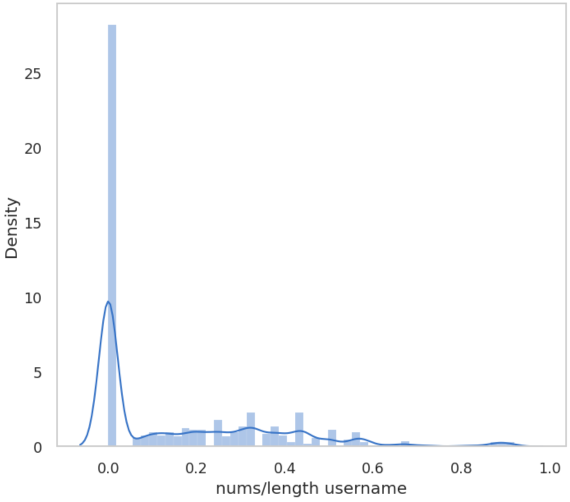

Fig:5(c) Data visualization data set (Fake profile)

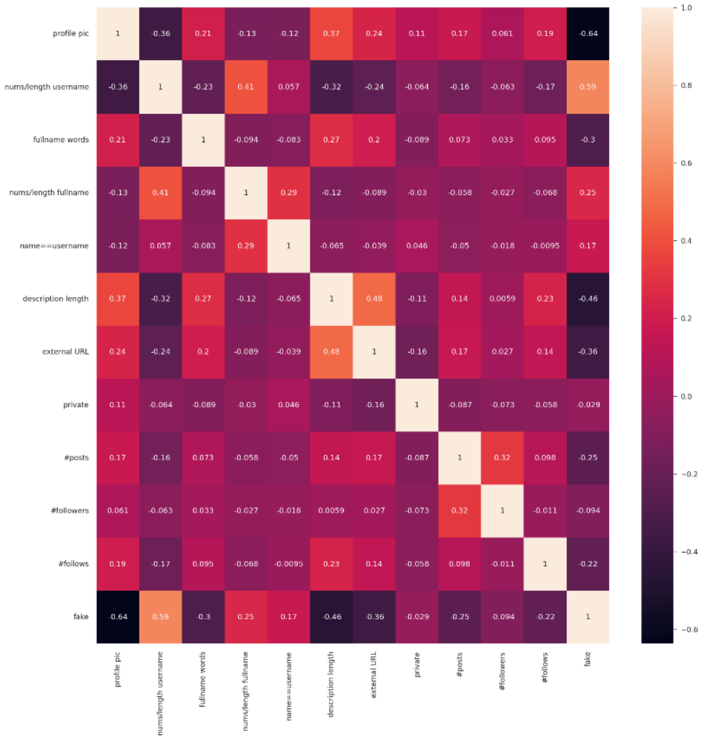

Fig:5(d) Data set information

1. **Feature engineering:** It will be done using binary conversion of features Conversion of certain features to binary values, such as nums/length fullname, private, and external URL. This transformation simplifies the model's interpretation and enhances its efficiency.

data['nums/length fullname'] = data['nums/length fullname'].apply(lambda x: 1 if x else 0)

data['private'] = data['private'].apply(lambda x: 1 if x else 0)

data['external URL'] = data['external URL'].apply(lambda x: 1 if pd.notna(x) else 0)

2. Once it's done split feature-targets, Separation of feature ('X) and the target variable ('y') from the data set. 'x' consists of selected numerical features, and 'y' represent each profiles is classified as fake or real.

X = data[['profile pic', 'nums/length username', 'fullname words', 'nums/length fullname', 'name==username',

'description length', 'external URL', 'private', '#posts', '#followers', '#follows', 'fake']]

y = data['fake']

3. **Data splitting** – division of the dataset into training ('x_train','y_train') and testing ('X-test','y_test') sets using 'train_test_split'. This partition ensures that the model is trained on the subset of data and evaluated on unseen data to assess generalization performance

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

6. RESULT ANALYSIS

This comprehensive evaluation reveals that a combination of higher learning rates and temperature scaling improves the overall performance of the Bi-LSTM model in detecting fake profiles on social platforms, after evaluating the performance of a **Bi-LSTM model** for detecting fake profiles across a range of learning rates and temperature-scaled sigmoid functions. The experiments involved assessing the model's accuracy, precision, recall, F1 score, and AUC (Area Under the ROC Curve) for 12 distinct configurations, combining three different learning rates (0.003, 0.005, 0.009) and four temperature values (0.1, 0.3, 0.5, 0.9).

Following table describes overall results obtained in 12 distinct configurations, First column contain value of Learning rate, second contains temperature variation, Third to seven all are result columns, Third shows accuracy, forth records Precision, Fifth contains recall values, Sixth col contains F1 Score , Seven column contains AUC values which comes 1 for all experiments.

Table 1. Overall results for (Learning Rate (0.3,0.5,0.9) and Temperature (0.1,0.3,0.5,0.9)

Learning Rate and Temperature						
Learning Rate	**Tempe-raptures**	**Accu-racy**	**Pre-cision**	**Recall**	**F1 Score**	**AUC**
0.003	0.1	0.896	0.827	1	0.906	1
0.003	0.3	0.903	0.837	1	0.911	1
0.003	0.5	0.906	0.842	1	0.914	1
0.003	0.9	0.911	0.849	1	0.918	1
0.005	0.1	0.927	0.872	1	0.932	1
0.005	0.3	0.932	0.88	1	0.937	1
0.005	0.5	0.932	0.88	1	0.937	1
0.005	0.9	0.935	0.885	1	0.939	1
0.009	0.1	0.957	0.921	1	0.959	1
0.009	0.3	0.957	0.921	1	0.959	1
0.009	0.5	0.963	0.93	1	0.964	1
0.009	0.9	0.966	0.936	1	0.967	1

Fig 6(a), Describes confusion matrix over the temperature(0.3,0.5,0.9 Fig 6(b) ROC curve of temperature (0.1) ,Fig 6(c)Accuracy ,Percision,Recall F1 Score for 0.1, Fig 6(d) Plots precision-recall curve, shows evaluating model with temperature over (0.3,0.5,0.9) on learning rate(0.003)

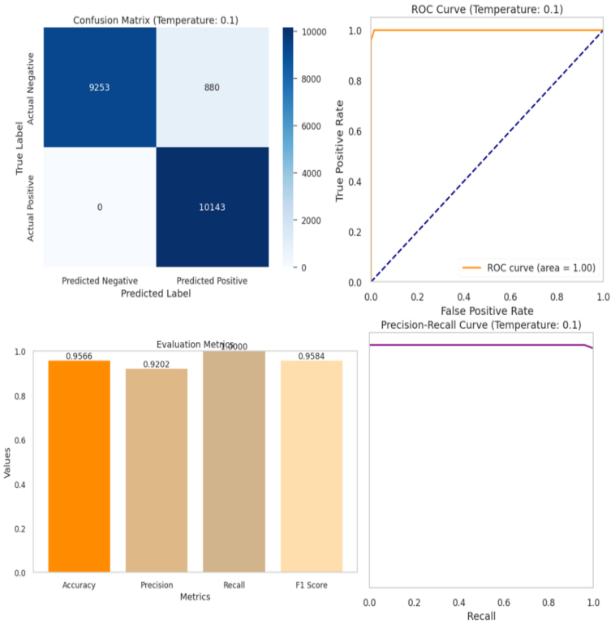

Fig:6(a) Evaluating model with temperature: 0.1 at learning rate: 0.003

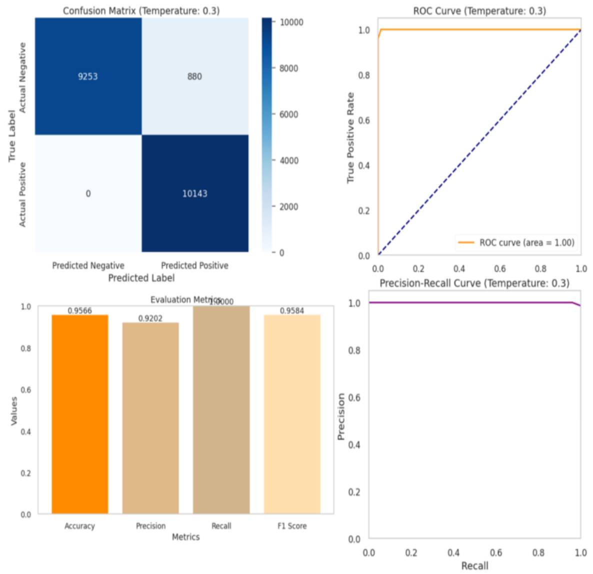

Fig:6(b)Evaluating model with temperature:0.3 learning rate:0.003

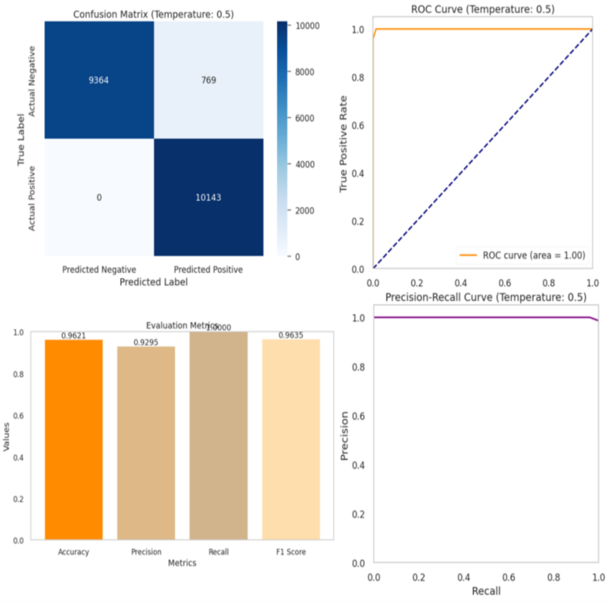

Fig:6(c) Evaluating model with temperature: 0.5 at learning rate: 0.003

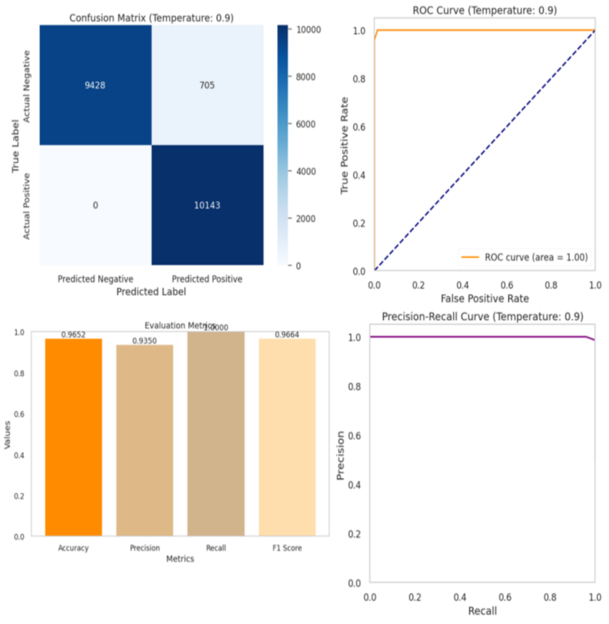

Fig:6(d) Evaluating model with temperature: 0.9 at learning rate: 0.003

Detail analysis on learning rate **0.003**:

1. At a learning rate of **0.003**, the model showed gradual improvement in performance as the temperature value increased from **0.1** to **0.9**.
2. **Accuracy** improved from **0.8951** at **0.1** temperature to **0.9103** at **0.9** temperature.
3. **Precision** increased steadily from **0.8267** at **0.1** to **0.8480** at **0.9**, indicating the model's growing confidence in correctly identifying true positive predictions.
4. **Recall** remained consistently at **1.0**, showing that the model was highly effective at identifying all actual fake profiles.
5. The **F1 Score** improved from **0.9051** to **0.9178**, suggesting a strong balance between precision and recall at higher temperatures.
6. **AUC** remained almost perfect at **0.99998**, highlighting the model's excellent ability to distinguish between fake and real profiles.

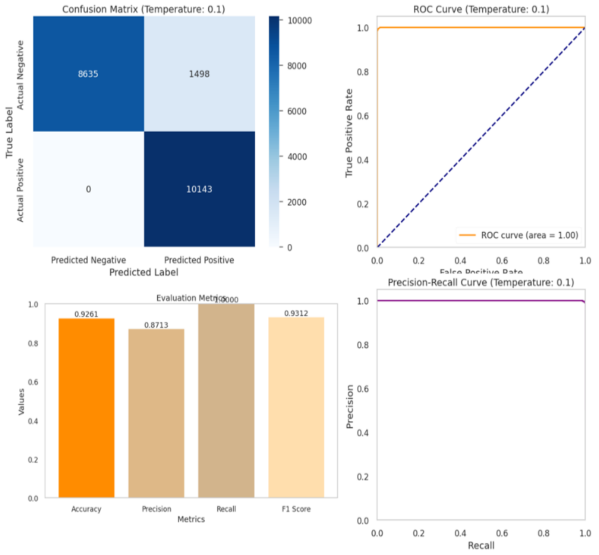

Fig:7(a) Evaluating model with temperature: 0.1 at learning rate: 0.005

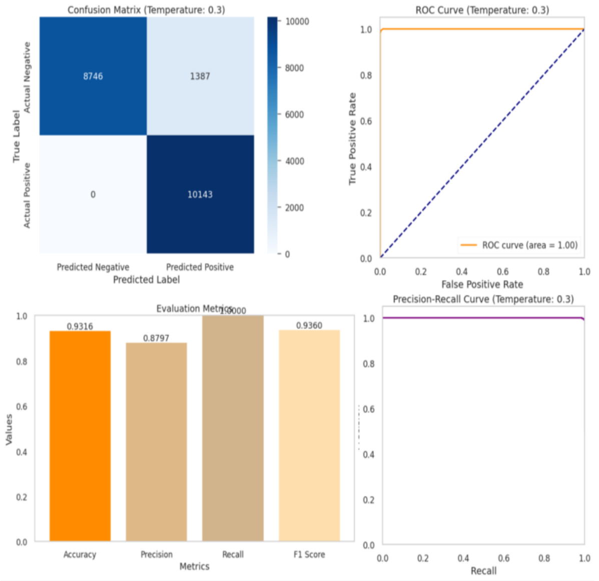

Fig:7(b) Evaluating model with temperature: 0.3 at learning rate: 0.005

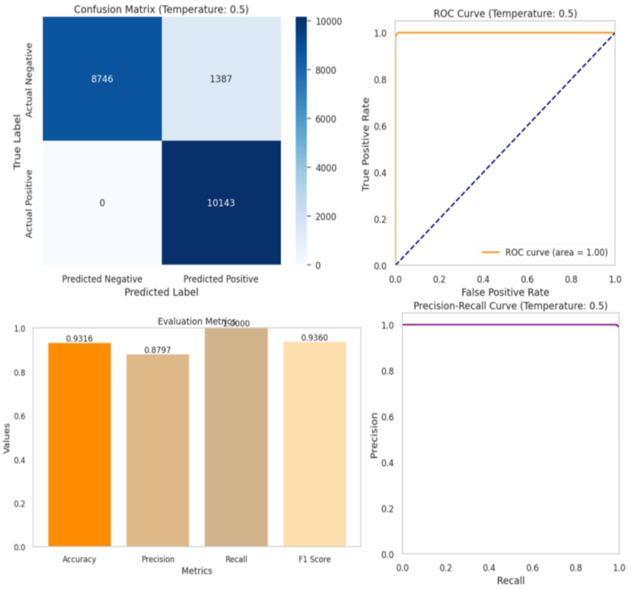

Fig:7(c) Evaluating model with temperature: 0.5 at learning rate: 0.005

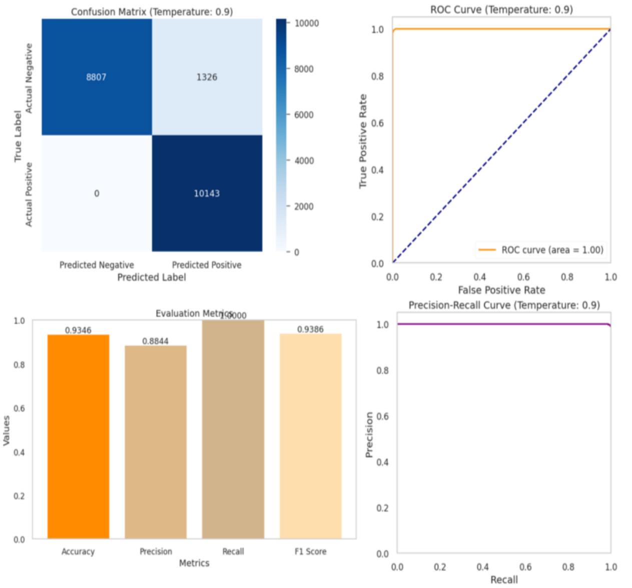

Fig:7(d) Evaluating model with temperature: 0.9 at learning rate: 0.005

Detail analysis on learning rate **0.005**:

1. At **0.005**, the model performance improved significantly across all metrics compared to **0.003**.
2. The **accuracy** increased to **0.9261** at **0.1** and reached **0.9346** at **0.9**, showing that the model became more accurate in classification as the temperature increased.
3. The **precision** also improved to **0.8713** at **0.1** and reached **0.8844** at **0.9**.
4. **Recall** remained consistently at **1.0** across all temperature values, meaning that the model continued to perfectly recall all fake profiles.
5. **F1 Score** followed the trend of accuracy and precision, increasing from **0.9312** to **0.9386**.
6. The **AUC** was slightly lower than the previous learning rate at **0.99991**, but still indicates excellent classification.

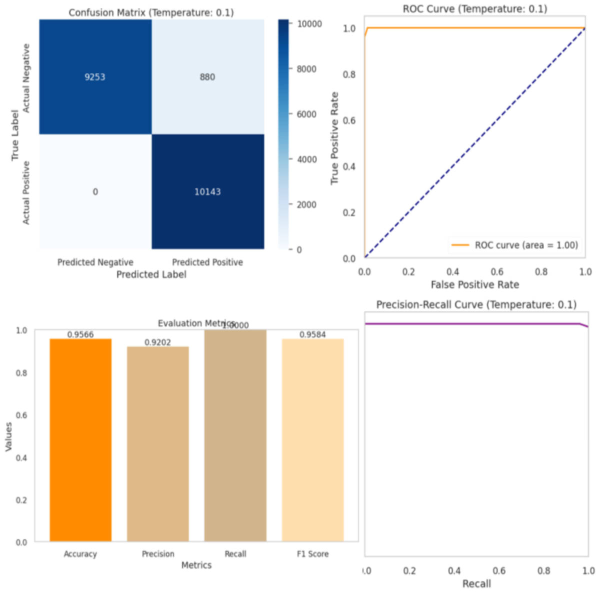

Fig:8(a) Evaluating model with temperature: 0.1 at learning rate: 0.009

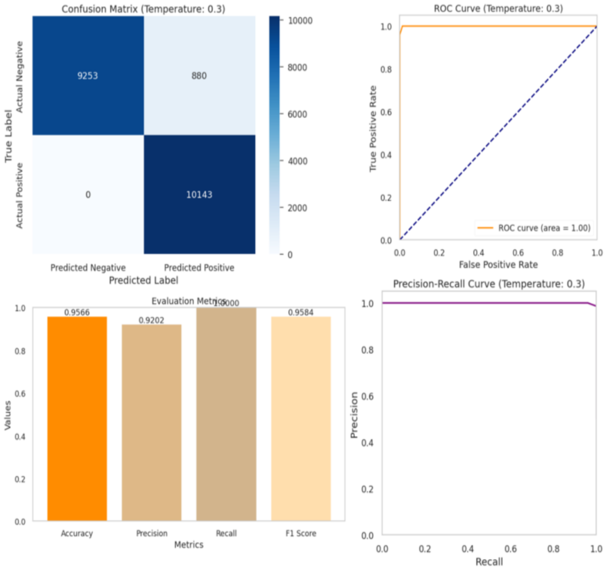

Fig:8(b) Evaluating model with temperature: 0.3 at learning rate: 0.009

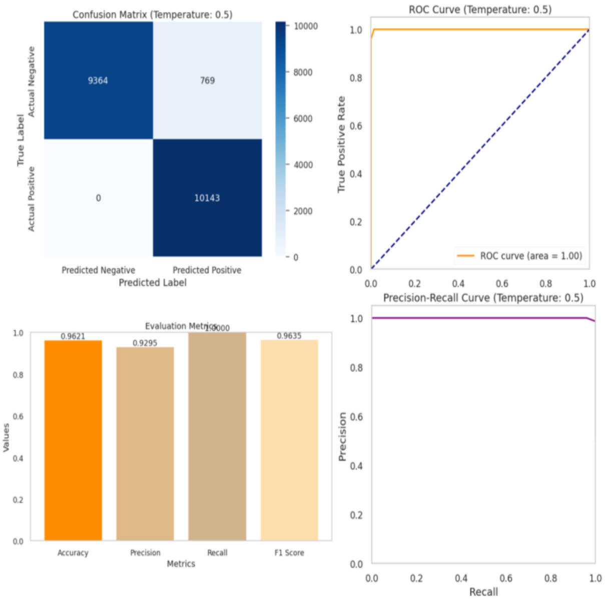

Fig:8(c) Evaluating model with temperature: 0.5 at learning rate: 0.009

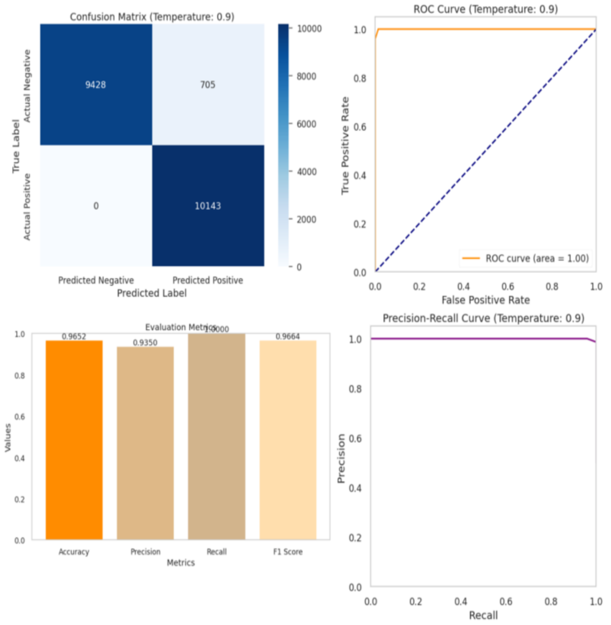

Fig:8(d) Evaluating model with temperature: 0.9 at learning rate: 0.009

Detail analysis of learning rate of **0.009**:

1. The highest learning rate of **0.009** produced the best overall results.
2. The **accuracy** increased significantly, reaching **0.9652** at **0.9** temperature, marking the highest performance in all configurations.
3. **Precision** also showed the best results at **0.9350**, signifying the model's ability to reduce false positives more effectively at higher learning rates.
4. **Recall** remained constant at **1.0**.
5. The **F1 Score** reached its peak at **0.9664**, demonstrating that both precision and recall were well-balanced at this configuration.
6. The **AUC** remained very high at **0.99972**, continuing to demonstrate excellent discrimination ability between the two classes.

7. RESULT DISCUSSION

The learning rate of 0.009 paired with a temperature of 0.9 provided the optimal results across all metrics, achieving the highest accuracy, precision, F1 score, and AUC values. The model performed consistently well across all temperature values, though higher temperatures, in combination with higher learning rates, generally yielded better precision and F1 scores. Recall remained 1.0 across all configurations, indicating that the model was highly effective at identifying fake profiles regardless of the learning rate or temperature settings. The AUC values were consistently near perfect, confirming the robustness of the model in distinguishing between real and fake profiles. Also able to answer all questions.

1. How can the effectiveness of Bi-LSTM networks be improved for detecting fake profiles on social platforms?

By leveraging temperature scaling in the softmax function, This paper aim to reduce overconfidence in the model's predictions. The research evaluates whether varying temperature values (0.1, 0.3, 0.5, 0.9) improves classification performance, and how this scaling impacts accuracy, precision, recall, F1-score, and AUC metrics.

2. How does varying learning rates influence the performance of the Bi-LSTM model?

The study explores the impact of different learning rates (0.003, 0.005, 0.009) on the convergence and stability of the model during training. This investigation helps determine the most suitable learning rate for balancing model performance and training efficiency.

3. What is the comparative impact of temperature scaling on different performance metrics?

By analyzing the performance across multiple temperatures, This paper aim to identify the optimal temperature setting that achieves the best balance between precision and recall, while minimizing false positives and false negatives.

4. Can temperature-scaled softmax provide more reliable probability estimates for fake profile detection?

Temperature scaling adjusts the confidence levels of softmax outputs. The research examines whether this adjustment leads to more reliable probability estimates, reducing the risk of misclassification in challenging datasets with high similarity between fake and genuine profiles.

5. How does the proposed method compare to traditional fake profile detection techniques?

The research compares the results of the Bi-LSTM model with temperature scaling against conventional approaches, highlighting improvements or drawbacks in detection accuracy and computational efficiency. This helps position the proposed approach within the broader context of existing methodologies

8. CONCLUSION

In conclusion, the optimized Bi-LSTM model with a learning rate of 0.009 and a temperature of 0.9 delivered the highest accuracy, precision, F1 score, and AUC, demonstrating its effectiveness in detecting fake profiles. The model consistently achieved perfect recall across all test scenarios, ensuring the reliable identification of fraudulent accounts. By incorporating temperature scaling within the softmax function, the model significantly improved the precision-recall balance while mitigating the risk of overconfident misclassifications. Additionally, the fine-tuning of learning rates played a crucial role in stabilizing the model and further enhancing its predictive performance. Compared to traditional detection methods, the proposed approach provides a robust, scalable, and efficient solution for safeguarding social media .While the current model has shown exceptional results, several avenues remain open for further enhancement. Future work will explore:

1. Expanding the input features to include image analysis, metadata patterns, and behavioral data alongside text-based classification.
2. Enhancing the model to handle multi-class categorization, enabling detection of not only fake vs. real profiles but also different types of fraudulent behaviors (e.g., bots, spam accounts, impersonation, etc.).
3. Implementing the model for real-time analysis to provide instant fake profile detection with minimal latency.
4. Implementing continual learning mechanisms that allow the model to adapt over time as fraudsters develop more sophisticated evasion tactics.

By focusing on these future directions, the proposed Bi-LSTM model can evolve into an even more intelligent, adaptable, and efficient framework for automated fraud detection in digital environments.

9. REFERENCES

[1] Guo, C., Pleiss, G., Sun, Y, 2017. Weinberger, K. Q, On Calibration of Modern Neural Networks

[2] Ferrara, E. 2016.The Rise of Social Bots.

[3] Subrahmanian, V.S 2016. The DARPA Twitter Bot Challenge.

[4] Kumar, S., et al. 2018. False Information on Web and Social Media: A Survey. Proceedings of the VLDB Endowment.

[5] Cresci, S., et al. 2015.Fame for Sale: Efficient Detection of Fake Twitter Followers. Decision Support Systems.

[6] Yang, Singh and Menczer 2024.Characteristics and prevalence of fake social media profiles with AI-generated faces

[7] Varol, O., et al. 2017. Online Human-Bot Interactions: Detection, Estimation, and Characterization.

[8] Almaatouq, A., et al. 2016. If It Looks Like a Spam Bot, and Behaves Like a Spam Bot, It Must Be a Spam Bot: Analysis and Detection of Microblogging Spam

[9] Sergio A. Balanya, Juan M, Daniel R 2024.Adaptive temperature scaling for Robust calibration of deep neural networks

[10] Andy S,Dorsa S,Stefano E 2023. Long Horizon Temperature Scaling

[11] Graves, A,Schmidhuber J 2005.Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks

[12] Schuster, M.,Paliwal, K. K. 1997. Bidirectional recurrent neural networks

[13] Wang, Z., Yan, W., & Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline

[14] Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. 2016. Character-aware neural language models.

[15] Cui, Z., Chen, W.,Chen, Y. 2016.Multi-scale convolutional neural networks for time series classification

[16] Karim, F., Majumdar, S., Darabi, H., & Harford, S. 2018, LSTM fully convolutional networks for time series classification

[17] Lipton, Z. C., Berkowitz, J., & Elkan, C. 2015,A critical review of recurrent neural networks for sequence learning

[18] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural Computation.

[19] Hashida, S., and Tamura, K. 2019. Multi-channel MHLF: LSTM-FCN using MACD-histogram with multi-channel input for time series classification.

[20] Mahara, G. S., and Gangele, S. 2021. A comprehensive survey of social network analysis-based anomaly detection techniques with soft computing.

[21] Ishlam, T. 2022. A proposed Bi-LSTM method for fake news detection. arXiv preprint.

[22] Mahara, G. S., and Gangele, S. 2022. A survey on detecting fake news in social media with AI: Challenges and possible directions.

[23] Mahara, G. S., and Gangele, S. 2022. Fake news detection: A RNN-LSTM, Bi-LSTM-based deep learning approach.

[24] Shu, K., Zhou, X., Wang, S., and Zafarani, R. 2019. The role of user profiles for fake news detection.

[25] Staudemeyer, R. C., and Morris, E. R. 2019. Understanding LSTM – a tutorial into Long Short-Term Memory recurrent neural networks.

[26] Cook, J., and Ramadas, V. 2020. When to consult precision-recall curves.

[27] Mao, A., Mohri, M., and Zhong, Y. 2023. Cross-entropy loss functions: Theoretical analysis and applications.

[28] Davis, J., and Goadrich, M. 2006. The relationship between precision-recall and ROC curves.