future internet

MDPI

Article

Countering Social Media Cybercrime Using Deep Learning: Instagram Fake Accounts Detection

Najla Alharbi, Bashayer Alkalifah, Ghaida Alqarawi and Murad A. Rassam *

Department of Information Technology, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia; 431214113@qu.edu.sa (N.A.); 431214115@qu.edu.sa (B.A.); 431214112@qu.edu.sa (G.A.)
* Correspondence: m.qasem@qu.edu.sa

Abstract: An online social media platform such as Instagram has become a popular communication channel that millions of people are using today. However, this media also becomes an avenue where fake accounts are used to inflate the number of followers on a targeted account. Fake accounts tend to alter the concepts of popularity and influence on the Instagram media platform and significantly impact the economy, politics, and society, which is considered cybercrime. This paper proposes a framework to classify fake and real accounts on Instagram based on a deep learning approach called the Long Short-Term Memory (LSTM) network. Experiments and comparisons with existing machine and deep learning frameworks demonstrate considerable improvement in the proposed framework. It achieved a detection accuracy of 97.42% and 94.21% on two publicly available Instagram datasets, with F-measure scores of 92.17% and 89.55%, respectively. Further experiments on the Twitter dataset reveal the effectiveness of the proposed framework by achieving an impressive accuracy rate of 99.42%.

Keywords: social media crime; Instagram fake accounts; deep learning; online social media; detection framework

check for updates

Citation: Alharbi, N.; Alkalifah, B.; Alqarawi, G.; Rassam, M.A. Countering Social Media Cybercrime Using Deep Learning: Instagram Fake Accounts Detection. *Future Internet* **2024**, *16*, 367. https://doi.org/10.3390/fi16100367

Academic Editor: Francesco Buccafurri

Received: 8 September 2024
Revised: 28 September 2024
Accepted: 8 October 2024
Published: 11 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The growth of technology has revolutionized how people communicate, interact, and do business. Online Social Networks (OSNs) sites like X and Instagram have revolutionized how people communicate in the last few years [1] "X" refers to the platform formerly known as Twitter. Various OSNs are currently available; a few of the most well-known networks are shown in Figure 1. These sites have become very popular in the 21st century.

Figure 1. Popular social networks.

This has made people spend a considerable amount of time online interacting with people and sharing various types of content, i.e., news, events, etc. [2].

Instagram is a platform that was initially designed to share photos and videos. The app enables users to instantly capture and share their moments either as a picture or a

video to friends. The app has more than 2 billion active users [3], and people have shared more than 40 billion photos since its launch in October 2010 [4]. Due to its versatility and ease of use, Instagram has become an ideal application for the proliferation of fake user accounts that behave in a very peculiar manner [5]. A fake account is mainly used to boost the Instagram metrics of accounts using fake followers. Fake accounts tend to improve the metrics of other Instagram users and thus create an unhealthy environment [6].

Instagram attacks are becoming a coordinated activity mainly associated with fake accounts [7]. Incorporating fake accounts in Instagram has been used to post fake bulk reviews to enhance product revenue or service [8]. Most of these users are involved in misleading and misinforming other genuine users. They are also involved in spreading malicious abuse, spamming, conning, and swaying the general public for their self-gain. Identifying these fake accounts becomes a significant task to protect honest users from these malicious and fake ones.

Instagram fake account detection is quite difficult for a variety of reasons. Current methods, such as machine learning algorithms, have limitations when it comes to accurately detecting and obtaining features from fake accounts. When working with complex, high-dimensional numerical data, these methods' significant reliance on manual feature engineering and selection makes detection challenging. By automatically identifying pertinent features from raw data, methods based on deep learning, on the other hand, have the potential to overcome the drawbacks of conventional machine learning techniques. While deep learning models have demonstrated high accuracy in various applications, their effectiveness depends on the complexity and size of the dataset. For smaller or more structured datasets, traditional machine learning methods may still perform better, making it essential to evaluate both approaches in the context of fake account detection.

While earlier research investigated the detection of fake accounts on platforms like Facebook and X, it must take into account Instagram's particular features, user habits, and data structures. Generalizing results from other platforms broadly to Instagram requires caution. Therefore, there is an intriguing research opportunity in this subject as there is a pressing need for more investigation to assess the usability and effectiveness of LSTM, especially for detecting fake accounts on Instagram.

The objectives of this research are as follows:

1. To build a strong and efficient fake account detection framework by leveraging the advantage of the power of LSTM networks. The major goal is to develop a system for detecting fake Instagram accounts, taking into account the particular characteristics and challenges presented by this social media environment.
2. To conduct a thorough performance comparison of the proposed LSTM-based framework with existing machine learning and deep learning algorithms typically used for fraudulent account identification. This objective entails determining the effectiveness and superiority of the LSTM technique in discriminating between real and fake Instagram accounts.
3. To validate the created LSTM-based framework's reliability and effectiveness by applying it to two separate Instagram datasets and one X dataset. This goal allows us to compare the results from the Instagram datasets to those from the X dataset, thus confirming the LSTM approach's usefulness in detecting fake accounts across several social media sites.

The contributions of this work are as follows: introducing a novel LSTM-based framework for detecting fake Instagram accounts and outperforming traditional machine learning models like Random Forest and CNN. It also validates the model's robustness and generalization across Instagram and X datasets, proving its effectiveness across different social media platforms.

The remainder of this paper is structured as follows: Section 2 presents a background on social spam and fake account detection domain. In Section 3, the related works are examined and analyzed. Section 4 explains the proposed framework's design and compo-

nents. Section 5 describes the results and analysis, providing a comprehensive discussion of the findings. Section 6 outlines the future research and concludes the paper.

2. Backgrounds

The background section defined the key concepts used in this research to recognize the essential aspects. Social spam was highlighted in all its forms. Moreover, most techniques used to detect fake accounts were highlighted, such as ML and DL.

2.1. Social Spam

With the growth of technology, new and sophisticated modes of spamming have come up. Extensive spamming involves a set of spammers working together like fake accounts, social spambots, or cloned accounts, etc., where they employ various spam propagation techniques to obtain information. In Figure 2, social spam categories are elaborated and spammer content and accounts are highlighted [9].

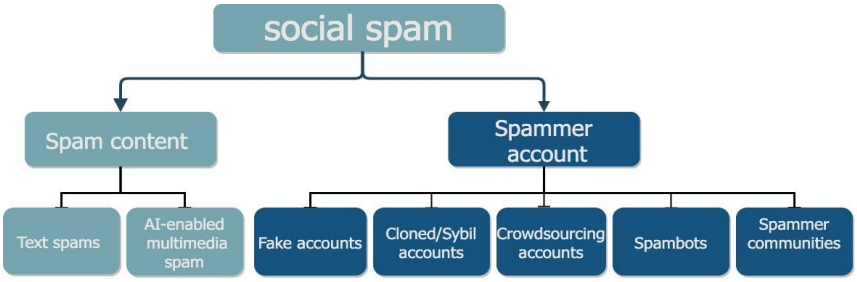

Figure 2. Social spam categories.

2.1.1. Social Spam Content

Unwanted spam content that appears on OSNs is known as social spam content. Social spam content is further divided into text spam and artificial intelligence (AI)-enabled multimedia spam. This can be further elaborated as follows:

1. Spam text is any text that is unwanted by a user's but however transmitted to others without their consent. They include a malicious link to open website, a link to download files, and a link sent to a phone number [10].
2. AI-enabled multimedia spam is a form of social spam that is AI-generated from a more extensive set of datasets to simulate and mimic human behavior, i.e., expression, voice, etc. [11].

2.1.2. Social Spam Content

Various forms of spammer account that do initiate and propagate spams exist in OSNs. During spamming, a legitimate user's account is not directly used, but rather attackers use groups of fake accounts, Sybil accounts, and spambots. This can be further elaborated as follows:

1. Fake accounts are legitimate user accounts controlled by cybercriminals to perform malicious activities on their social network sites. They are mainly used to send fake friend requests, spread fake information, and spread malicious content to target unaware victims. With the growth of technology, detecting fake accounts is increasingly tricky. Fake accounts have mastered the way of mimicking legitimate users in OSNs [8].
2. Cloned/Sybil accounts refer to multiple fake user identities that mainly work by providing the OSN with fake and malicious data. The Sybil users may contain a list of accounts mainly made up of the same users or have similar intentions to many users [12].
3. A crowdsourcing account is a form of marketing strategy where an organization works by outsourcing its online, operational services to an ambiguous group of

people or another company. This may include freelancers for commercial services, spamming, swaying followers in social media accounts, etc. However, astroturfing is a marketing strategy used to create a false impression of widespread support of something. However, an organization behind this marketing strategy conceals sit identity [13].

4. Spambots are controlled computer programs that mimic human activity to spread spam content across the internet. They automatically produce content on social media and operate at a significantly higher pace. Spambots are hard to detect since they keep their robotic identity and nature undisclosed. An example of the adoption of spambots was the US midterm election in 2010, where political communication was manipulated by spreading malicious posts to websites with fake news headlines [14].
5. A spammer community refers to an organized crime syndicate where social spammers collaborate and form a collective anomaly to spread spam messages to legitimate users. The spammers collaborate with other spammers and working in tandem to influence and control their effectiveness online to form a social spammer community [8].

2.2. Fake Account Detection

Fake profile accounts are mainly used to obtain information illegitimately, defame, and spread malware [15]. AI technology allows computers systems to mimic human behavior and intelligence. Most fake accounts detection mechanisms are mainly feature-based or graphical-based to identify fake identities [16]. This technology includes machine learning (ML) and a deep learning (DL) technique that allows the machine to access user experience to improve task management [17], as shown in Figure 3.

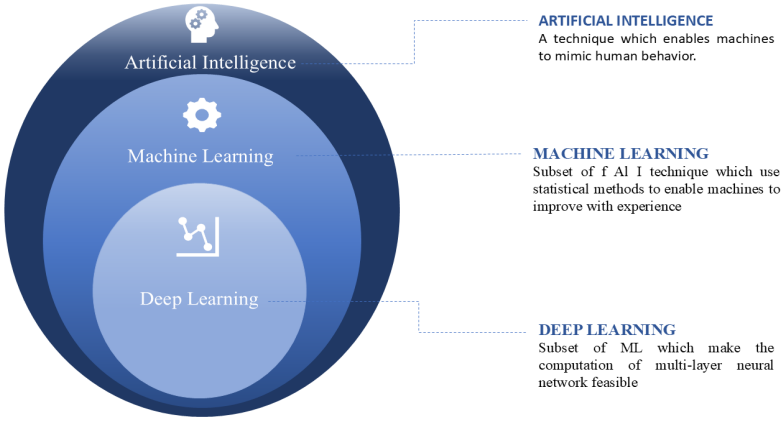

Figure 3. DL as subfield of AI [18].

At present, fake account detection is one of the most critical challenges for OSNs. ML and DL currently play an essential role in fake account detection in OSNs.

2.2.1. Machine Learning

Machine learning refers to the capability of machines to mimic intelligent human behavior [19]. ML can be categorized into two key categories, i.e., supervised ML and unsupervised ML [20].

1. Supervised ML a classification-based model is utilized to categorize data [21]. It has a wide range of available algorithms, such as support vector machines (SVM), decision trees (DT), Random Forest (RF), and artificial neural network (ANN), each with its strengths and weaknesses [22].
2. Unsupervised ML is a form of ML that deals with grouping unlabeled data into various clusters [21]. This form of ML is mainly applied in data mining to cluster data patterns into similar groups instead of giving the prediction directly. They are thus used as complementary tools to supervised ML [20].

2.2.2. Deep Learning

Unlike other traditional data classification models requiring handcrafted features to be used to perform modeling classification, DL learn these features by analyzing the layers that are hidden in data. DL features must be able to progressively learn and interpret high-level data features [23].

A Recurrent Neural Network (RNN) is a form of network created for the purpose of sequential data modeling. It is mainly used in the modeling and processing of natural language. A standard RNN only deals with a limited variant's length. Therefore, to solve the problem of the dependence in RNN, different RNN versions have been designed. They include LSTM, bi-RNN, and a gated recurrent unit [24].

Long Short-Term Memory (LSTM) is based on time series data. LSTM is a variant of RNN that is mainly used for classification, processing, and prediction [25]. It helps to solve the vanishing gradient problem, which is mainly encountered in the RNN. It circumvents units by employing multiple switch gates. It usually contains memory cells that receive the input and the previous state [26].

In general, it can be concluded that the LSTM cell is mainly composed of an input layer, an output layer, and a self-connected hidden layer. The cells work by choosing what to keep in memory and what to discard. To form the following input, the cells join the current output with the previous state. Thus, they can capture the term dependencies [27].

This network is mainly preferred due to: the following factors:

1. It can take a sequence of information and uses the recurrent mechanisms and gate techniques.
2. It uses the feedback gained to remember previous states.

3. Related Works

In recent years, fake accounts have become a serious problem in OSNs. Currently, many researchers have conducted studies on detecting fake accounts on different social platforms.

A study in [26] suggested using a trained model of a deep neural network along with specific parameters to identify automatic spam and fake Instagram account profiles. The model for identifying fake Instagram profiles was prepared using the gradient descent technique. The gradient descent was also used to obtain the optimized network loss and bias values. The cost function was minimized iteratively. The proposed method's precision and accuracy were 93% and 91%, respectively. The emphasis on precision over recall reflects their strategy to minimize false positives while detecting fake Instagram accounts.

The researchers in [28] proposed a framework for detecting fake profiles on Facebook. This framework would be based on a deep neural network (DNN) classifier. The results showed good efficiency and performance, with an accuracy of 99.4%.

This study [29] developed a detection model that uses a range of machine learning approaches to discriminate between real and fake X profiles. Based on features including follower and friend count and status changes, it distinguished between real and fake accounts using the X profile dataset. Moreover, it made use of neural networks, XG Boost, Random Forest, and LSTM. The features are selected to evaluate the veracity of a social media profile; the result of this research showed that XG Boost has the best performance out of the machine learning techniques for finding fake profiles, with 99.6% accuracy.

The study in [30] covers a variety of machine learning algorithms, including ensemble approaches (such as Random Forest and XGBoost), support vector machines (SVM), decision trees, Bagging, and Boosting. This research seeks to identify a method for distinguishing between real and fake Instagram profiles based on publicly available profile data. The Synthetic Minority Oversampling Technique (SMOTE) technique was used to make the two data classes equal. This makes it easier to obtain the same number of instances for each class. The results demonstrate an overall highest accuracy of 96% for Random Forest and XGBoost.

In the study of [31], a hybrid method based on SMOTE incorporating Differential Evolution (DE) was proposed to facilitate real-time spam rate detection in X datasets. The

study was mainly aimed to help in identifying real owners and to filter the spam posts. Incorporating SMOTE would help tackle the imbalance of class distribution in the datasets, while the DE will help tackle the RF hyperparameters. The results of the hybrid method increased accuracy from 89.4% to 99.3%.

Another study in [32] proposed a fake account detection framework for effectively classifying an account as true or fake on X. This framework would be based on a logistic regression (LR) classifier. The results showed good efficiency and performance, with an accuracy of 96.2%.

A study conducted in [33] proposed incorporating a technique involving AI technology and Natural Language Processing and a way to increase the accuracy level in fake account detection. The two techniques worked by incorporating a RF Classifier, optimized Naive Bayes (NB) technique that help to profiles category, and SVM. The three algorithms' techniques were incorporated to determine the false and right identities of the accounts. These algorithms use fewer features and have a very high accuracy rate of successfully detecting fake profiles.

The study conducted in [34] proposed a supervised ML model that would help to classify and authenticate fake users. The research conducted aimed to identify fake users' behavior. The dataset used contained fake users and authentic users. The research incorporated 17 features to be used. The RF algorithm demonstrates high accuracy in both 2-class classification (authentic and fake users) and 4-class classification (authentic users, active fake users, inactive fake users, and spammers). The results obtained showed that the RF algorithm had an accuracy level of 91.76%. Furthermore, it revealed a difference between fake and authentic users.

The researchers in [35] introduced different methods for detecting spam accounts on X based on classification and clustering methods. The methods used SVM, Multi-Layer Perceptron (MLP), and RF. RF methods have better results in performance, with an accuracy of 96.2%.

A study conducted in [36] proposed identifying the minimal profile data required to identify fake profiles in the LinkedIn platform. The research also proposes an appropriate data mining technique for identifying fake profiles. The techniques use four key data mining ways, i.e., neural networks, SVM, Weighted Average (WA) for a fake profile, and Principal Component Analysis (PAC). The technique incorporated showed an increase in the approximate accuracy by 14%. Collectively, it led to an increase in the accuracy rate of fake profiles on LinkedIn.

The study [37] proposed a Deep Profile technique that would incorporate a deep neural network algorithm that would help detect and control the fake accounts in the social media network. The research used a dynamic convolutional neural network (CNN) learning model to classify a fake profile. In addition, it proposed the inclusion of a novel pooling that would help to optimize and enhance performance of neural network's during the training process. The results obtained show a higher accuracy rate with a relatively lower loss compared to common learning algorithms.

A study [38] proposed a graphical-based semi-supervised learning algorithm (EGSLA) that would help detect fake users from a large pool of large X data. The proposed method included four main modules: the collection of data, feature extraction, classification of data, and decision-making techniques. The results of the study show that the EGSLA achieved 90.3% accuracy in detecting fake accounts.

The authors in [39] proposed a technique to detect malicious and fake user accounts by putting a Chrome extension as a service on a user's homepage on the X platform. The technique is based on ML and graph-based trace and aims to differentiate fake accounts from real ones. The experimental results showed excellent performance, with an accuracy of 99.16%.

The study conducted in [6] involved detecting fake and automated accounts that would help detect fake engagement in Instagram accounts. In this research, two key datasets were generated. ML algorithms like LR and neural networks were used. To detect

automated accounts in Instagram, a cost-sensitive algorithm technique was applied due to its unnatural bias in the dataset. From the research conducted, automated account detection yielded an accuracy of 86%, and fake account detection yielded an accuracy of 96%.

In the study [16], the authors proposed a model to detect account and message spam on X based on ML. The authors use behavioral and content-based features for spammer detection. The results showed that RF had the best accuracy for detecting spam messages, and LogitBoost (LB) had the best accuracy for detecting fake accounts.

The study [40] proposed to incorporate a K-means algorithm that integrates a levy flight using a sinusoidal map to facilitate the tuning of the generated absorption coefficient produced. The study primarily aimed to model the spammers on X using a bioinspired computational method by including 13 modeling factors. The results obtained had an accuracy of 97.98% and a fast convergence rate.

The study [41] presents a novel approach for detecting bots on social media platforms, specifically X. It uses a deep learning model based on LSTM architecture, which combines post content analysis with metadata. The model is trained on a dataset of around 3000 X bot examples. The model achieves high classification accuracy, with an accuracy score exceeding 96% for bot detection at the post level. When applied to account-level detection, the accuracy increases to 99%, indicating its effectiveness in distinguishing between bots and human users.

The study in [42] proposed to incorporate a message tree that works by identification of the internal relations that exist amongst the hidden suspicious accounts. This study focused more on the forwarded messages of the internal relations than on suspicious accounts. The result of the proposed gives very good accuracy and false positive rates (95.3% and 0.5%, respectively).

The study [5] used ML to reveal spam accounts on X. This study builds a dataset of spam accounts, and it was publicly available. The results showed an accuracy of 97.5% for RF, 90.4% for Decorate, 90.4% for DT, 76.7% for Adaptive Boosting (AB), 89.1% for Bayesian Network (BN), 94.6% for k-Nearest Neighbors (KNN), 55.1% for LR, and 95.5% for SVM.

Discussion

The existing literature on fake account detection has investigated various methods, including the use of ML algorithms such as RF Classifier, SVM, NB, PCA, and LR. Table 1 summarizes existing fake accounts detection studies. Specifically, some research has used ML approaches such as hybrid ML, J48 Decision Tree, and ANN to analyze Instagram data. Based on the different studies conducted in this field, ML has a significant role in detecting fake accounts. However, one major disadvantage of ML algorithms is the fact that they rely on manual feature engineering and selection, which requires domain expertise and can be time-consuming. This constraint is particularly challenging when dealing with high-dimensional numerical data, considering that determining significant features is a diligent task.

In contrast, other research has used deep learning techniques such as LSTM and dynamic CNN to navigate these challenges. DL distinguishes itself from typical ML by automatically extracting and learning the most relevant features from raw data using a layered architecture. This included feature extraction capacity provides a substantial advantage for handling complicated high-dimensional datasets without requiring heavy user intervention.

Furthermore, deep learning, particularly LSTM, is well suited to addressing the challenges presented by sequential data, such as text. It performs exceptionally well in tasks that involve processing data with inherent temporal or sequential dependencies, as in the case of email spam detection. For example, studies in [43,44] have effectively used LSTM to analyze large datasets for spam email detection, where the sequential nature of the data is a key factor in achieving accurate results.

Although LSTMs are traditionally applied to sequential data, this research explores their adaptability to non-sequential datasets. LSTM's ability to capture complex rela-

tionships between features, even in non-sequential contexts, makes it a candidate worth investigating. This study aims to determine whether LSTM can outperform traditional architectures, such as CNNs and classical ML algorithms, in detecting fake Instagram accounts. To achieve this, multiple architectures will be tested and compared to assess LSTM's effectiveness in this novel application. Furthermore, the inclusion of dropout layers addresses the risk of overfitting, ensuring the model generalizes well to unseen data. While most research has focused on platforms like X and Facebook, Instagram's unique features warrant further exploration to assess the applicability of LSTM in this context. This investigation addresses a gap in the current literature and provides an opportunity to evaluate LSTM's potential for detecting fake accounts on Instagram compared to other architectures.

Table 1. Summary of the literature.

Ref.	Year	OSN	ML/DL	Techniques Used	Accuracy (A)	Limitation
[45]	2023	Instagram	Hybrid ML	Neural network	A = 93%	Low accuracy
[28]	2023	Facebook	DL	DNN	A = 99.4%	It focuses on a Facebook dataset
[29]	2022	X	DL	LSTM	A = 99.6%	No combination multiple models used for more better performance
[30]	2022	Instagram	ML	XGBoost and RF	A = 96.00%	One dataset
[31]	2021	X	Hybrid ML	LR, DE, and RF	A = 99.9%	The datasets used did not have a relationship amongst the individual and thus made it impossible to draw a social graph
[32]	2021	X	ML	LR	A = 96.2%	LR is vulnerable to over-fitting and can handle only small volumes of data
[33]	2021	OSNs	Hybrid ML	RF Classifier, SVM, and Optimized NB	A = 97.6%	The research requires manual feature selection
[34]	2020	Instagram	ML	RF, ANN, LR, NB, and J48 Decision Tree	Up to A = 91.76%	The computation time is enormous in supervised learning Unwanted data reduces efficiency
[35]	2020	X	ML/DL	RF, SVM, and MLP	Up to A = 96.2%	SVM performance reduces with an increase in data size MLP is requires a large number of iterations and it only for linear data
[36]	2020	LinkedIn	ML	SVM, PCA, and WA	A = 87.34%	Limited profile data
[37]	2020	OSNs	DL	Dynamic CNN	A = 94%	Limitation of neuron number It enlarges computing resource
[38]	2019	X	ML	EGSLA	A = 90.3%	Iteration results are not stable It has low accuracy
[39]	2019	X	Hybrid ML	RF, Bagging, JRip, Random tree, PART, J48, and LR	A = 99.16%	The RF is that with an increase in tress it leads to a decrease in the algorithm performance and it has high computational cost and slow prediction generator
[6]	2019	Instagram	ML/Dl	NB, LR, ANN, and SVM	Up to F-Measure = 94%	LR is prone to over-fitting SVM is not suitable for large and complex datasets
[16]	2019	X	ML	LB	A = 97.7%	difficult or complex to work with a large amount of data
[40]	2018	X	Hybrid ML	K-means integrated levy flight algorithm	A = 97.98%	K-means algorithm cannot handle noisy data and outliers
[41]	2018	X	DL	LSTM	A = 99%	It focuses on an X dataset
[42]	2017	Sina Weibo	ML	Message tree	A = 95.3%	Data collection for the study of OSN was rendered very difficult due to personal private issues A small variation in data can lead to a greater change in the structure of the decision tress thus leading to instability
[5]	2015	X	ML	RF, Decorate, DT, AB, BN, KNN, LR, and SVM.	Up to A = 97.5%,	Lack of parallelism, slow computation, and ineffective memory utilization

4. Proposed Framework

The framework is divided into five stages, with each stage linked to the next stage. In the first and second stage, the input and data are preprocessed before developing the framework. In the next stage, the LSTM framework is described in detail. In the fourth stage, the results are critically discussed. In the last stage, the proposed framework is compared with other model to prove its efficiency and effectiveness. The conceptual of framework is shown in Figure 4.

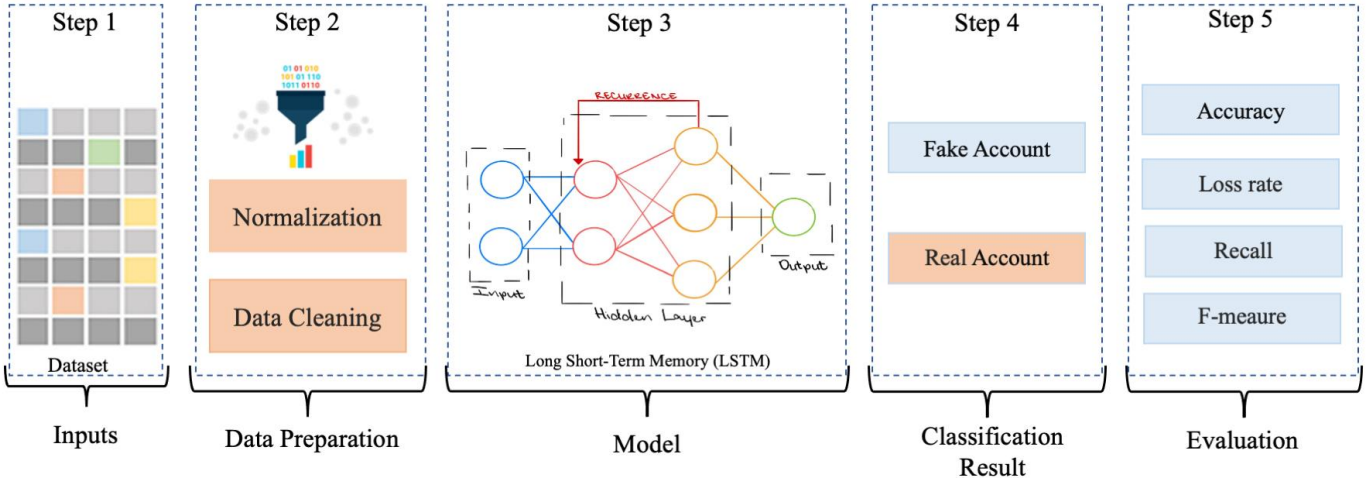

Figure 4. Conceptual framework.

4.1. Stage1: Input (Dataset and Features)

Two datasets were used to detect fake Instagram accounts, which are described in more detail below, along with an X dataset to enhance the analysis further and improve the accuracy of the detection process.

4.1.1. Dataset 1

The study used the "InstaFake" dataset offered by GitHub [46]. According to the data, there are 1002 actual accounts and 201 fake accounts. This includes accounts from other countries and fields. Eight features have been paid close attention to. This includes things like follower and following counts, media counts, media posting dates or frequency, media comments, followed and following accounts, profile picture of the account, and username of the account profile. The list of user features used for the detection framework is explained in Table 2.

Table 2. List of user features in dataset 1.

Description	Feature Name
user_media_count	Total number of posts an account has.
userFollowerCount	The account's follower count.
userFollowingCount	Following of account
user_has_profil_pic	Whether or not an account has a profile picture.
user_is_private	Whether an account is a private profile or not.
userBiographyLength	The total number of characters in the account biography.
username_length	The number of characters in the username of the account.
username_digit_count	The number of digits in the username of the account.
is_fake	True, if account is a spam/fake account, False otherwise

4.1.2. Dataset 2

The study used "Instagram fake spammer s accounts" dataset offered by Kaggle [47]. There were 696 users in the selected dataset. The dataset comprises an exact balance

of 348 real users and 348 fake users, resulting in a meticulously balanced database. The dataset contained 10 features such as profile picture, username, number of followers, and number of accounts the user is following. The list of user features used for detection is explained in Table 3.

Table 3. List of user features in dataset 2.

Description	Feature Name
profile Pic	Whether the account have profile pic or not.
nums/Length Username	The percentage of a username's length to its number of numeric characters.
fullname Words	Whole name in word tokens.
nums/Length Fullname	The percentage of the number of numeric characters in whole name to its length.
name = =Username	Username and whole name are equals.
description Length	User bio length in characters.
external URL (Uniform Resource Locator)	Whether the account have an external c or not.
Private	Whether the account is private or public.
Posts	Total posts number.
Followers	Follower count of the account.
Fake	True, if account is a spam/fake account, False otherwise.

4.1.3. Dataset 3

The third dataset is a Twitter (now "X") dataset with 3474 real users and 4912 social spam accounts. There are 69 features defined for each of these accounts. These features can be categorized into three groups: elements that relate to content, account information, or account usage. The dataset is available for academic research online under the name "Dataset for supervised bot detection on Twitter" [40]. Although Instagram operates differently from other social networks like Facebook or X, the dataset used in this study comes from X due to the similar features shared between the platforms in detecting fake accounts. Additionally, the dataset from X is the best available option for this type of analysis, providing a robust basis for evaluation.

4.2. Stage2: Data Preparation

In this stage, two sub-stages are used for data preparation, that is, normalization and cleaning.

4.2.1. Normalization

Data normalization is rescaling one or more variables to a range between 0 and 1. The goal of the normalization of data is to convert the numeric values in to a similar scale in a dataset without losing the data or distorting the ranges of values.

4.2.2. Data Cleaning

Data cleaning is the process of removing extraneous or erroneous information from data before it can be analyzed. This is data that, by propagating a false notion, can have a negative impact on the framework or algorithm into which it is given. The common data cleaning criteria include resolving missing values and deleting duplicates.

4.3. Stage3: Framework Structure

The LSTM technique will be used at this stage, and the information that has been pre-processed will be entered. In order to extract the output, the framework will process data using LSTM. LSTM is a special type of recurrent RNN. It refers to the series of neural networks that works by processing sequential data. LSTM has three gates, as shown in Figure 5. The three "gates" are serially arranged in an LSTM unit. It has an input gate, forgetting gate, and output gate. When a piece of data is added to the LSTM network, it is

carefully selected by the rules. Information that adheres to the algorithm rules is the only one left out. The rest is sent to the forgetting gate.

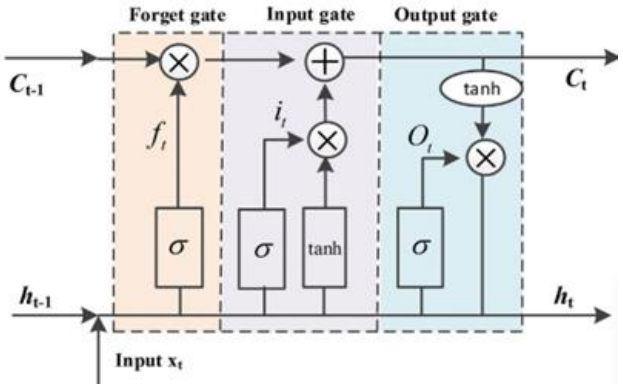

Figure 5. LSTM cell structure [48].

The work of the gates is to allow the information to be passed selectively. The gates allow passage of information selectively in an ordering manner. Equation (1) illustrates the LSTM network's default activation functionality and the sigmoid function. The gating unit has the capability of adding or deleting information.

$$\text{sigmoid layer } \sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

The sigmoid neural network layer and the multiplication pair operation helps to determine whether the information passes or not. The sigmoid layer consists of an element output that is represented by real numbers between (0.1) that represent the weight of the information passing.

The LSTM neural network consist of a layer that has the function of tanh activation. This is shown in Equation (2). The tanh is used for updating the state of neurons Equation (2).

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2}$$

The LSTM cell's information flow, as shown in Figure 2, can mathematically be described. The symbols $\oplus$ and $\otimes$ denote the adding and multiplying value of the framework, respectively, while the arrows denote information direction of flow. The first layer of the memory gate helps determine how to remove any information that is not necessary from the cell state. This can mathematically be written as

$$f_t = \sigma(W_f \times x_t + U_f \times h_{t-1} + b_f) \tag{3}$$

where f_t represents the forgetting threshold at time t, and σ represents the sigmoid activation function. W_f and U_f represents the weights, x_t. represents the input value h_{t-1} and represents the output value at time $t-1$, and b_f represents the bias term.

The function of the second input gate is to determine which specific information from the current input vector should be saved in the cell state. Then, it assists in updating the cell state value, and the tanh layer helps generate the new value state of C_t. The following expressions help to further elaborate:

$$i_t = \sigma(W_i \times x_t + U_i \times h_{t-1} + b_i) \tag{4}$$

$$\overline{C_t} = \sigma(W_c \times x_t + U_c \times h_{t-1} + b_c) \tag{5}$$

These expressions represent the input gate at time (t); the weights W_i, U_i, W_c, and Uc; and the bias factors (b_c) and b_i. The following expressions are used to update the state of the cell at a given time:

$$C_t = f_t \times C_{t-1} + i_t \times \overline{C_t} \quad (6)$$

The output information in the current time step is primarily produced in the third layer. It is written as

$$O_t = \sigma(W_o \times x_t + U_o \times h_{t-1} + b_o) \quad (7)$$

The output threshold at time t is denoted by O_t, the weights are W_o and U_o, and the bias term is b_o. The cell's output value can then be written as follows:

$$h_t = O_t \times \tanh(C_t) \quad (8)$$

where h_t represents the cell's output value at time t, tanh represents the activation function, and C_t represents the cell's state at time t. When data are sent through the gates, the important information is labeled as output, while the invalid data are labeled as forgotten [48].

LSTM Parameters and Performance Measures

The framework uses a series of hyperparameters that help to enhance the results. Five parameters, the optimizer, batch size, epochs, loss function, and learning rate, are employed and summarized in Table 4 to assess the performance of the proposed framework. There are numerous optimizers that could distinguish between different frameworks; some of these optimizers are displayed in Table 5.

Table 4. Parameters of LSTM.

Parameters	Description
Optimizer	Refers to adjusting the framework parameters to minimize the framework error in the training step.
Loss function:	Used to optimize your framework. It is the function that is minimized by the optimizer.
Batch Size	The number of data samples that are sent via the network before the parameters are updated.
Epoch	Refers to an iterative process of the training framework that occurs at each iteration
Learning Rate	A hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated

Table 5. LSTM Optimizers.

Optimizers	Description
RMSProp	RMSProd effectively eliminates the influence of historical gradient by substituting an exponentially decaying moving average for the summation of squared gradient in AdaGrad [49].
Adaptive Moment Estimation (Adam)	Adam is a learning method created exclusively for deep neural network training. Adam has the advantages of being more memory efficient and using less compute resources [50].
AdaGrad	With its more complex AdaGrad algorithm, learning rates are scaled inversely in relation to the square root of the cumulative squared gradient [50].
Stochastic gradient descent (SGD)	Due to SGD's frequent updates and large variation, the objective function highly fluctuates [51].

4.4. Stage 4: Classification Result

The dataset's results will be displayed at this stage, and it will be determined whether the account is real or fake. The obtained results will then be analyzed in the next stage.

4.5. Stage 5: Evaluation

The Evaluation stage highlights a detailed assessment of the proposed framework. Several metrics and parameters helped to evaluate the efficiency and effectiveness of the algorithms used. This parameter can be elaborated as follows:

True positive (TP): This represents the ratings that were initially classified as positive and later predicted to be positive by the classifier.

False positive (FP): This represents the ratings that were classified as negative and later predicted to be positive by the classifier.

True negative (TN): This represents the ratings that were initially classified as negative and were also predicted to be negative by the classifier.

False negative (FN): This represents the ratings that were initially classified as positive but were predicted to be negative by the classifier.

Various parameters were used to evaluate the proposed classifier performance [52]. They include the following:

Accuracy refers to level of agreement between actual and absolute measurements. This metric describes the proportion of categorized samples to the total number of samples and is used to classify performance. It is expressed as

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \tag{9}$$

Loss Rate refers to the function that quantifies the difference between the actual and predicted output during training in order to improve the learning process. It aids in the evaluation of model performance and is primarily concerned with minimizing error. It is written as

$$\text{Loss} = -\text{Y} \times \text{Log}(\text{YPred}) - (1 - \text{Y}) \times \text{Log}(1 - \text{YPred}) \tag{10}$$

Recall refers to the capacity of a classification model to recognize all of the data points in a class. It aids in the calculation of the number of samples retrieved divided by the total number of correct samples. It is expressed as

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{11}$$

F-Measure precision and recall are used to calculate the F-measure. It depicts how both measures behave. It is expressed as

$$\text{F} - \text{Measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

4.6. Framework Setup

This experiment was carried out using Python library packages such as Pandas, Numpy, Matplotlib, and Keras on Google CoLab. LSTM cells, drop out, and dense output layers were used to construct the fake account detection system. Dropout layers were specifically included to mitigate the overfitting problem. The layers and parameter values are described in Table 6. The framework also used the Adam optimizer and sigmoid activation functions. The framework was run with 50 and 100 epochs and a batch size of 25.

Table 6. Framework setup settings.

Parameters	Value
Language	Python
Libraries	Pandas, Numpy, Matplotlib, and Keras
Train set	70%
Test set	30%
Activation functions	sigmoid activation
Optimizer	Adam
Epochs number	50 and 100
Batch size	25

5. Results and Analysis

This section examines the results of applying the proposed framework. In addition, a review of the results is provided.

5.1. Experimental Results

The proposed framework performance and accuracy are tested using various optimizers and modifications. In two Instagram datasets, the evaluation has employed several optimizers, each of which yields a different result, including Adagrad, Adam, RMSprop, and SGD. The "Adam" Optimizer provides the best performance, as shown in Tables 7 and 8.

Table 7. Dataset 1 improvement results with 100 iterations.

Dataset 1			
Optimizer	**Accuracy (%)**	**Loss Rate**	**Time (s)**
Adam	97.42%	0.085	147 s
RMSprop	96.81%	0.075	200 s
Adagrad	82.33%	0.726	120 s
SGD	82.33%	0.700	100 s

Table 8. Dataset 2 improvement results with 100 iterations.

Dataset 2			
Optimizer	**Accuracy (%)**	**Loss Rate**	**Time (s)**
Adam	94.21%	0.144	12 s
RMSprop	92.59%	0.1693	25 s
Adagrad	79.93%	0.6737	13 s
SGD	86.44%	0.376	12 s

To assess the framework's performance, the proposed framework is tested on two datasets with various iterations (50 and 100). In 150 s, the suggested framework for dataset 1 achieved 97.14 percent accuracy, 0.097 loss rate, 94.64% percent recall, 89.83% percent precision, and 91.37% percent F-measure using 50 iterations, according to Table 9. The framework exhibits some progress in accuracy to 97.42% percent after 100 iterations, with a loss rate of 0.085. In Table 10, the framework performed well for dataset 2 at 50 and 100 iterations. Figures 6 and 7 show the accuracy and loss rate for a specific iteration with a different database, respectively.

Table 9. Iterative results of dataset 1 utilizing the Adam optimizer.

Dataset 1						
Iterations	**Accuracy (%)**	**Loss Rate**	**Recall (%)**	**Precision (%)**	**F-Measure (%)**	**Time (s)**
50	97.14%	0.097	94.64%	89.83%	91.37%	150 s
100	97.42%	0.085	94.64%	89.65%	92.17%	228 s

Table 10. Iterative results of dataset 2 utilizing the Adam optimizer.

Dataset 2						
Iterations	**Accuracy (%)**	**Loss Rate**	**Recall (%)**	**Precision (%)**	**F-Measure (%)**	**Time (s)**
50	92.22%	0.188	92.30%	92.42%	92.30%	15 s
100	94.21%	0.144	92.30%	96.72%	89.55%	12 s

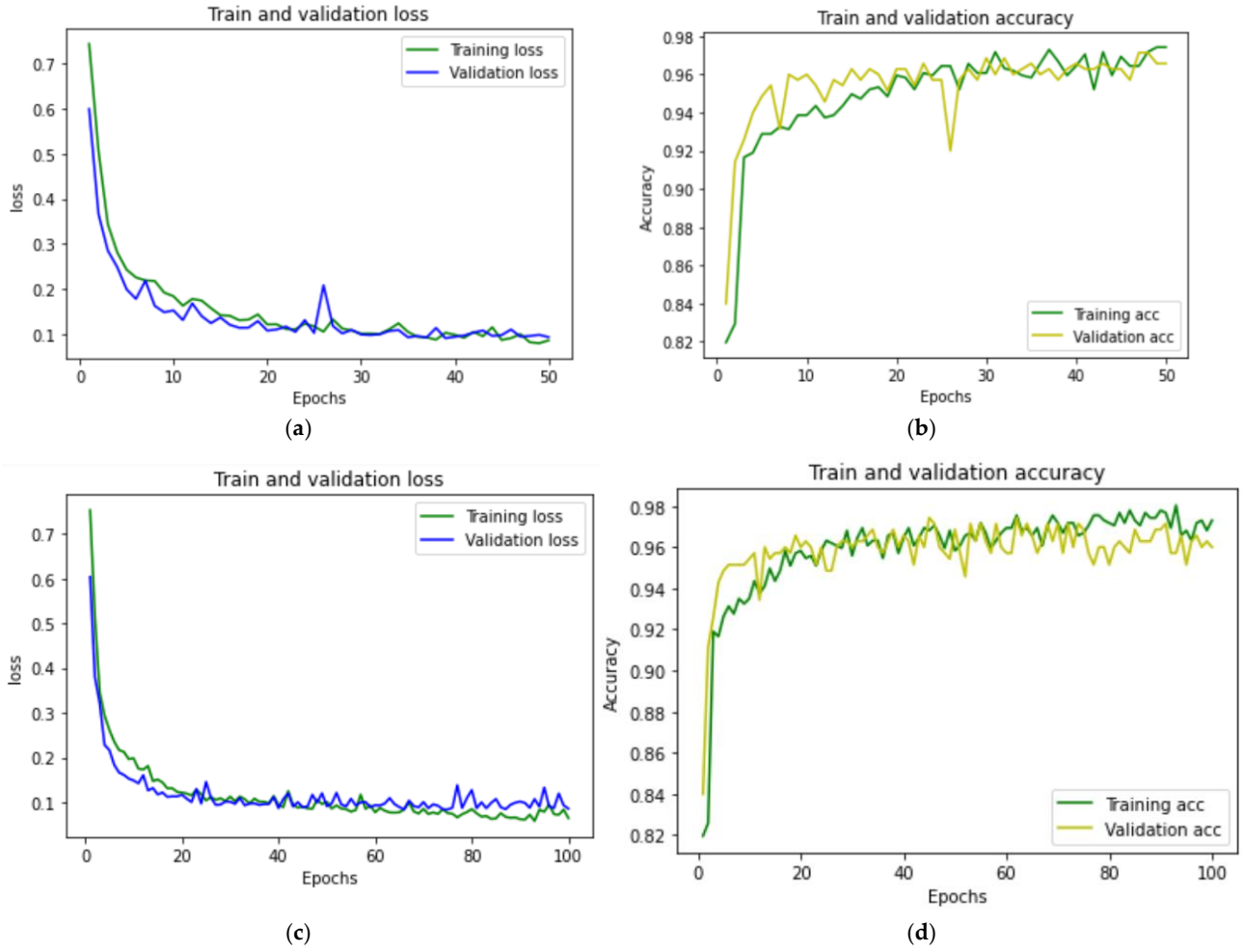

Figure 6. Results on dataset 1: (**a**) shows 50 iterations loss, (**b**) shows 50 iterations of accuracy, (**c**) shows 100 iterations lost, and (**d**) shows 100 iterations of accuracy.

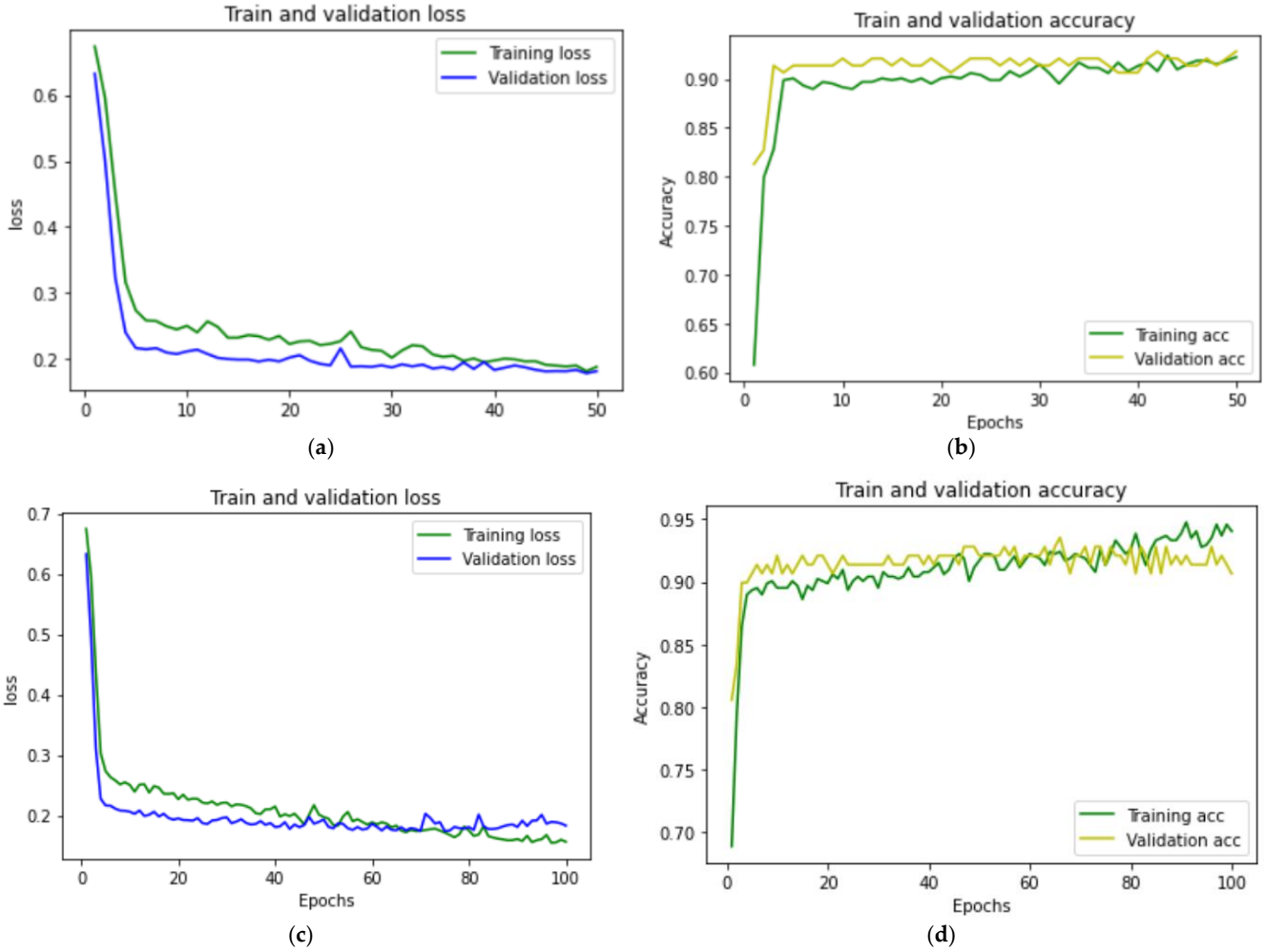

Figure 7. Results on dataset 2: (**a**) shows 50 iterations loss, (**b**) shows 50 iterations of accuracy, (**c**) showes 100 iterations lost, and (**d**) shows 100 iterations of accuracy.

5.2. Results Analysis

By using LSTM cells, the framework efficiently and effectively detects fake accounts. Many optimizers can be utilized in this framework; nevertheless, picking the best optimizer improves the results significantly. In experiment dataset 1, the increase in iterations has a marginal impact on accuracy, loss rate, and precision but notable affects F-measure. Meanwhile, in experiment dataset 2, the increase in iterations notably affects accuracy, loss rate, precision, and F-measure, with recall remaining unaffected.

The results showed a significant relationship between the loss rate and the number of iterations, as seen in Figures 6 and 7, showing that the framework is learned with each iteration. Furthermore, there is a positive relationship between time and the number of iterations; as the number of iterations increases, so does the duration.

5.3. Comparison with Machine Learning- and Deep Learning-Based Techniques

A comparison was made between the ML techniques; the comparison aimed to evaluate the performance of the proposed framework. The comparison involved two datasets. The comparison involved the proposed framework against other existing ML models such as GaussianNB (GNB), LR, RF classifier, and SVM. Tables 11 and 12 present a comprehensive evaluation of various ML frameworks, highlighting the performance of the proposed framework. Remarkably, the proposed framework exhibited the best overall performance among the considered frameworks. However, it is noteworthy that the RF classifier demonstrated higher accuracy, specifically in database 1. Due to the structured and non-sequential

nature of dataset 1, it plays to the strengths of the RF algorithm, which can handle such data types efficiently and requires less data for training. Nevertheless, despite its notable results, it can encounter challenges when handling enormous datasets. As the data size grows, the computational and memory requirements of training the RF model can become significant. This can lead to increased training times and resource constraints. On the other hand, the comparative analysis conducted in this study involved the proposed framework and other existing DL models, namely MLP, CNN, and DNN classifiers.

Table 11. Comparison of ML and DL results with the proposed framework on dataset 1.

Classifier		Accuracy	Recall	F-Measure
ML	SVM	93.31%	72.2%	76.5%
	GNB	91.92%	66.7%	71.3%
	LR	94.75%	74.1%	80.8%
	RF	97.77%	87.0%	92.2%
	KNC	95.26%	87.0%	84.7%
	DT	96.93%	90.7%	89.9%
	LDA	92.75%	59.3%	71.1%
DL	DNN	97.21%	87.03%	90.38%
	Dynamic CNN	95.82%	77.77%	84.84%
	MLP	96.10%	81.48%	86.27%
	Proposed framework	97.42%	94.64%	92.17%

Table 12. Comparison of ML and DL results with the proposed framework on dataset 2.

Classifier		Accuracy	Recall	F-Measure
ML	SVM	50.71%	100%	65.6%
	GNB	67.46%	98.0%	73.8%
	LR	88.99%	91.8%	88.7%
	RF	93.30%	90.8%	91.3%
	KNC	90.43%	88.8%	89.7%
	DT	89.95%	88.8%	82.2%
	LDA	88.51%	81.6%	87.0%
DL	DNN	91.42%	89.47%	91.89%
	Dynamic CNN	90.71%	89.47%	91.27%
	MLP	90.71%	88.15%	91.15%
	Proposed framework	94.21%	92.30%	89.55%

Notably, the LSTM-based model showcased distinct performance metrics across two distinct datasets. In the first dataset, the LSTM model outperformed DNN, dynamic CNN, and MLP in terms of accuracy, recall, and F-measure, clearly demonstrating its superior capability. However, in the second dataset, although our model achieved higher accuracy and recall rates, DNN, dynamic CNN, and MLP exhibited better F-measure scores. This discrepancy suggests that, while our model excelled in overall accuracy and capturing true positives and negatives, the competing models demonstrated a stronger balance between precision and recall, yielding higher F-measure scores.

It is noteworthy to a discuss related study [28] on Facebook fake profile detection, which indicated that a DNN model outperformed LSTM and Bidirectional LSTM (BiLSTM) models. However, in the instance of Instagram, this research demonstrates that the LSTM model outperforms the DNN. This discrepancy emphasizes the necessity of platform-specific analysis when comparing the effectiveness of various deep learning models for detecting fake accounts.

To summarize, the research effectively developed an LSTM-based framework for detecting fake Instagram accounts; when compared to ML and DL methods, they demonstrate

higher accuracy and recall. This framework effectively used LSTM's capabilities in automatically extracting relevant features to meet the limitations of manual feature engineering. A comparative investigation of two Instagram datasets confirmed the model's effectiveness. This study extremely advances fake account detection techniques by providing a robust solution adapted to Instagram's unique environment. To our knowledge, there are three known approaches, and an analysis of those methods is shown in Table 13. The variability in datasets can significantly affect the reported accuracy, as different data characteristics (such as size, class distribution, and feature richness) can lead to variations in model performance. As a result, the accuracy comparisons across methods may not provide a fully reliable assessment without considering the impact of the dataset differences.

Table 13. Comparison of approaches for fake account detection on Instagram.

Ref	Approach	Accuracy	Dataset Source
[45]	Neural Network	A = 91%	576 Instagram accounts: 288 were fake and 288 were genuine.
[34]	RF, ANN, LR, NB, and J48 Decision Tree	A = 91.76%	32,869 fake users and 32,460 authentic users, totaling 65,329 users on Instagram.
[6]	NB, LR, ANN, and SVM	A = 86%	1203 Instagram accounts, 201 were fake, and 1002 were genuine.
Proposed Method	LSTM	A = 97.42%	Two datasets Dataset 1: 1203 Instagram accounts, 201 were fake and 1002 were genuine. Dataset 2: 696 Instagram users, 348 were real users and 348 were fake users.

To improve the reliability of such comparisons, future work should ideally use a standardized dataset across all models to provide more consistent and reliable evaluations of performance.

5.4. Apply LSTM on X Dataset

In addition to the primary testing conducted on the two Instagram datasets, this study incorporated the X database as an additional evaluation step to further validate the performance of the proposed model. While the Instagram database served as the main testbed due to its relevance to the research objectives, the inclusion of the X database provided an opportunity to assess the generalizability and robustness of the model across different social media platforms. This multi-platform evaluation approach enhances the reliability and comprehensiveness of the findings by contributing to knowledge advancement in social media analytics.

The framework effectively detects X bot accounts by utilizing LSTM cells. The optimal selection of the optimizer significantly enhances the results. In this experiment, the RMSProp optimizer demonstrated the best performance. Table 14 shows the investigation in dataset 3. The study analyzed the impact of increasing iterations on key performance metrics, including accuracy, recall, loss rate, precision, and F-measure.

Table 14. Iterative results of dataset 3 utilizing the RMSProp optimizer.

	Dataset 3					
Iterations	**Accuracy (%)**	**Loss Rate**	**Recall (%)**	**Precision (%)**	**F-Measure (%)**	**Time (s)**
50	99.00%	0.0300	98.54%	99.51%	99.13%	19 s
100	99.44%	0.0279	99.10%	99.73%	99.52%	43 s

The increase in iterations has a marginal impact on accuracy recall, loss rate, precision, and F-measure. The results showed a significant relationship between the loss rate and the

number of iterations, as seen in Figure 8, showing that the framework is learned with each iteration. Furthermore, there is a positive relationship between time and the number of iterations; as the number of iterations increases, so does the duration.

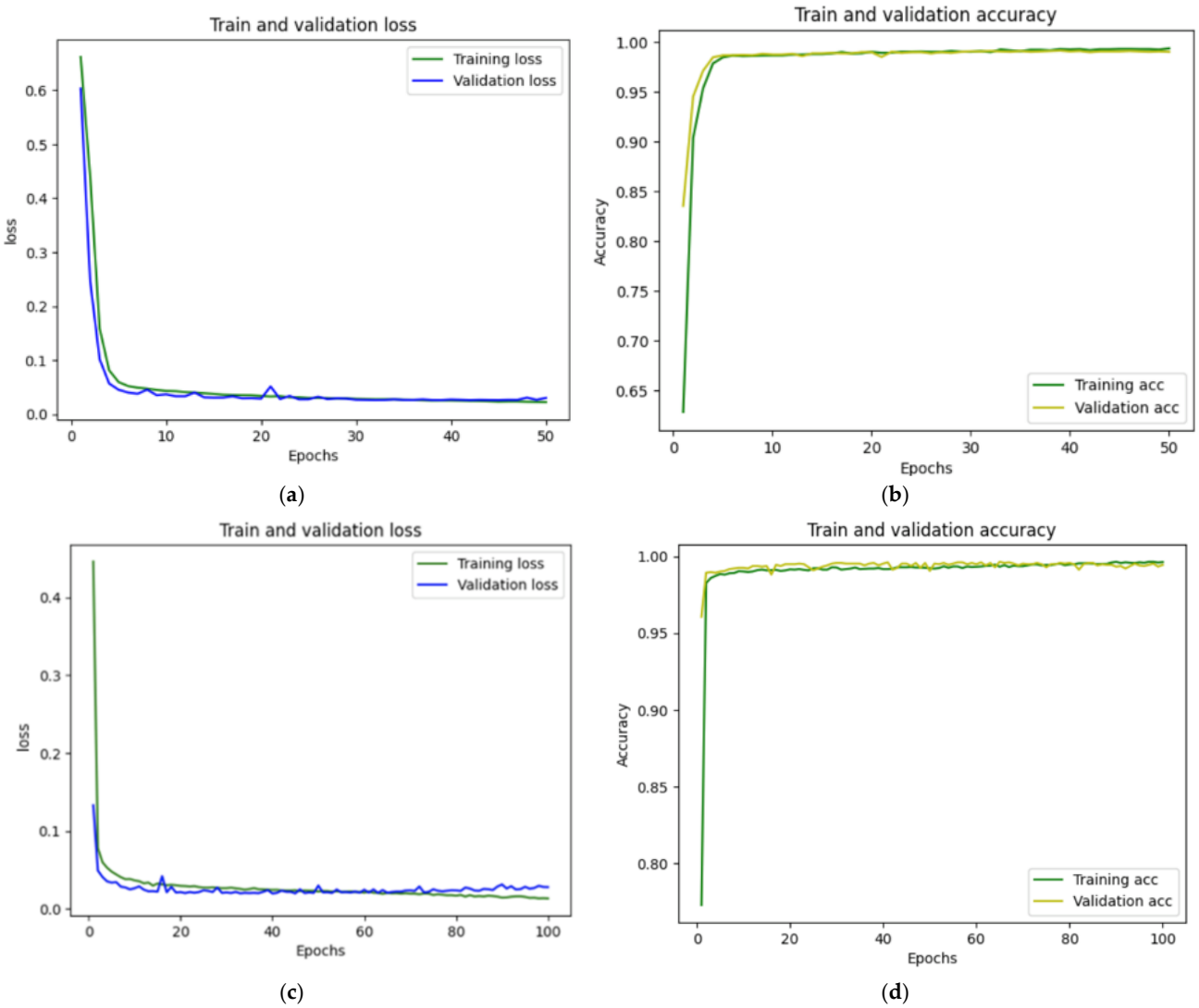

Figure 8. Results on dataset 3: (**a**) shows 50 iterations loss, (**b**) shows 50 iterations of accuracy, (**c**) showes 100 iterations lost, and (**d**) shows 100 iterations of accuracy.

5.5. Comparison with Other Machine Learning- and Deep Learning-Based Techniques

Applying the LSTM-based model to the X dataset produced remarkable results that demonstrated the power of the framework, as demonstrated in Table 15. The results of the investigation clearly show that the LSTM model is more accurate than the other ML methods currently in use, as demonstrated by its excellent F-measure scores. It is noteworthy, nevertheless, that the recall performance of the LR and GNB models was higher than that of the LSTM model. These findings suggest that the LSTM model excels in overall accuracy and effectively captures true negatives and true positives, as demonstrated by its high F-measure scores. However, as evidenced by their superior ability to detect true positives, the LR and GNB models indicate how effective they are at reducing false negatives and enhancing recall. Furthermore, an X dataset was used to compare the performance of the LSTM model with the Dynamic CNN and DNN models. The results demonstrated that the Dynamic CNN and DNN models outperformed LSTM in each of the performance criteria. This indicates that Dynamic CNN and DNN models, which are specifically designed to handle the particular issues given by X data, have more robust

feature extraction and pattern identification skills. Consequently, these findings show the need to choose models that are consistent with the intrinsic properties of the data, thereby identifying possible domains for further development and investigation within the LSTM model to improve performance in the realm of social media analytics.

Table 15. Comparison of ML results with proposed framework of dataset.

	Classifier	Accuracy	Recall	F-Measure
ML	SVM	77.50%	72.50%	78.80%
	GNB	92.88%	99.50%	94.10%
	LR	96.22%	99.40%	96.80%
	RF	99.16%	99.00%	99.30%
	KNC	93.08%	93.80%	94.00%
	DT	99.00%	99.10%	99.10%
	LDA	98.72%	98.80%	98.90%
DL	Proposed framework	99.44%	99.12%	99.52%

6. Conclusions and Future Work

This paper introduces the first utilization of LSTM in Instagram for fake account detection. The experiments conducted using the real dataset for the Instagram accounts show that the proposed framework effectively and efficiently detects fake accounts. The performance evaluation and comparison with other ML models showed the superiority of the proposed framework, achieving an accuracy of 97.42% for the first dataset and 94.21% for the second dataset.

Furthermore, the framework underwent comprehensive testing using an X dataset, resulting in an impressive accuracy of 99.42%. Including the X database in the evaluation enabled the assessment of the model's generalizability and robustness across diverse social media platforms and yielded valuable insights into its performance. These results collectively underscore the robustness and versatility of the proposed model, positioning it as a promising solution for detecting fake accounts across various social media platforms.

Building upon these promising outcomes, several avenues exist for future research and development in fake account detection on social media platforms. While our framework exhibited remarkable accuracy on Instagram and X data, it would be beneficial to extend its assessment to a broader spectrum of social media platforms, each with its unique user behaviors and characteristics. Finally, investigating the potential integration of additional data sources, such as user interactions, geographic information, and content analysis, could provide richer contextual information for improved accuracy.

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

Funding: This research received no external funding.

Data Availability Statement: Data are available at https://github.com/BashayerAlkalifah/Fake-Instagram-Account-Detection-/tree/main (accessed on 7 April 2024).

Acknowledgments: The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2024-9/1).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Çıtlak, O.; Dörterler, M.; Doğru, İ.A. A survey on detecting spam accounts on Twitter network. *Soc. Netw. Anal. Min.* **2019**, *9*, 1–13. [CrossRef]
2. Alom, Z.; Carminati, B.; Ferrari, E. A deep learning model for Twitter spam detection. *Online Soc. Netw. Media* **2020**, *18*, 100079. [CrossRef]
3. Roberts, J.A.; David, M.E. Instagram and TikTok Flow States and Their Association with Psychological Well-Being. *Cyberpsychology Behav. Soc. Netw.* **2023**, *26*, 80–89. [CrossRef] [PubMed]
4. Karayiğit, H.; İnan Acı, Ç.; Akdağlı, A. Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods. *Expert Syst. Appl.* **2021**, *174*, 114802. [CrossRef]
5. Cresci, S.; di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Fame for sale: Efficient detection of fake Twitter followers. *Decis. Support Syst.* **2015**, *80*, 56–71. [CrossRef]
6. Akyon, F.C.; Kalfaoglu, M.E. Instagram Fake and Automated Account Detection. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019, Izmir, Turkey, 31 October–2 November 2019. [CrossRef]
7. El-Mawass, N.; Honeine, P.; Vercouter, L. SimilCatch: Enhanced social spammers detection on Twitter using Markov Random Fields. *Inf. Process. Manag.* **2020**, *57*, 102317. [CrossRef]
8. Rao, S.; Verma, A.K.; Bhatia, T. A review on social spam detection: Challenges, open issues, and future directions. *Expert Syst. Appl.* **2021**, *186*, 115742. [CrossRef]
9. Ferrara, E. The history of digital spam. *Commun. ACM* **2019**, *62*, 82–91. [CrossRef]
10. Yang, H.; Liu, Q.; Zhou, S.; Luo, Y. A spam filtering method based on multi-modal fusion. *Appl. Sci.* **2019**, *9*, 1152. [CrossRef]
11. Fagni, T.; Falchi, F.; Gambini, M.; Martella, A.; Tesconi, M. TweepFake: About detecting deepfake tweets. *PLoS ONE* **2021**, *16*, e0251415. [CrossRef]
12. Li, Y.; Chang, M.-C.; Lyu, S. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, 11–13 December 2018; pp. 1–7. [CrossRef]
13. Cruickshank, I.J.; Carley, K.M. Characterizing communities of hashtag usage on Twitter during the 2020 COVID-19 pandemic by multi-view clustering. *Appl. Netw. Sci.* **2020**, *5*, 66. [CrossRef] [PubMed]
14. Ferrara, E. Measuring social spam and the effect of bots on information diffusion in social media. In *Complex Spreading Phenomena in Social Systems. Computational Social Sciences*; Springer: Cham, Switzerland, 2018. [CrossRef]
15. Sahoo, S.R.; Gupta, B. Fake Profile Detection in Multimedia Big Data on Online Social Networks. 2020. Available online: https://scholar.google.com/scholar?q=Sahoo,%20S.R.;%20Gupta,%20B.%20Fake%20profile%20detection%20in%20multimedia%20big%20data%20on%20online%20social%20networks,%202020 (accessed on 7 October 2024).
16. Adewole, K.S.; Anuar, N.B.; Kamsin, A.; Sangaiah, A.K. SMSAD: A framework for spam message and spam account detection. *Multimedia Tools Appl.* **2017**, *78*, 3925–3960. [CrossRef]
17. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
18. ODSC Team. Artificial Intelligence and Machine Learning in Practice: Anomaly Detection in Army ERP Data. 2019. Available online: https://opendatascience.com/artificial-intelligence-and-machine-learning-in-practice-anomaly-detection-in-army-erp-data/ (accessed on 9 October 2024).
19. Sohail, A.; Arif, F. Supervised and unsupervised algorithms for bioinformatics and data science. *Prog. Biophys. Mol. Biol.* **2019**, *151*, 14–22. [CrossRef]
20. Hood, S.B.; Cracknell, M.J.; Gazley, M.F. Linking protolith rocks to altered equivalents by combining unsupervised and supervised machine learning. *J. Geochem. Explor.* **2018**, *186*, 270–280. [CrossRef]
21. Soheily-Khah, S.; Marteau, P.-F.; Bechet, N. Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the iscx dataset. In Proceedings of the 2018 1st International Conference on Data Intelligence and Security, ICDIS 2018, South Padre Island, TX, USA, 8–10 April 2018; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018; pp. 219–226. [CrossRef]
22. Bao, W.; Lianju, N.; Yue, K. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. *Expert Syst. Appl.* **2019**, *128*, 301–315. [CrossRef]
23. Jain, G.; Sharma, M.; Agarwal, B. Optimizing semantic LSTM for spam detection. *Int. J. Inf. Technol.* **2019**, *11*, 239–250. [CrossRef]
24. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
25. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
26. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst. Appl.* **2021**, *185*, 115524. [CrossRef]
27. Smagulova, K.; James, A.P. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Spéc. Topics* **2019**, *228*, 2313–2324. [CrossRef]
28. Amankeldin, D.; Kurmangaziyeva, L.; Mailybayeva, A.; Glazyrina, N.; Zhumadillayeva, A.; Karasheva, N. Deep Neural Network for Detecting Fake Profiles in Social Networks. *Comput. Syst. Sci. Eng.* **2023**, *47*, 1091–1108. [CrossRef]
29. Chakraborty, P.; Shazan, M.M.; Nahid, M.; Ahmed, K.; Talukder, P.C. Fake Profile Detection Using Machine Learning Techniques. *J. Comput. Commun.* **2022**, *10*, 74–87. [CrossRef]

30. Sallah, A.; Alaoui, A.A.E.; Agoujil, S.; Nayyar, A. Machine Learning Interpretability to Detect Fake Accounts in Instagram. *Int. J. Inf. Secur. Priv.* **2022**, *16*, 1–25. [CrossRef]
31. Abkenar, S.B.; Mahdipour, E.; Jameii, S.M.; Kashani, M.H. A hybrid classification method for Twitter spam detection based on differential evolution and random forest. *Concurr. Comput.* **2021**, *33*, e6381. [CrossRef]
32. Bharti, K.K.; Pandey, S. Fake account detection in Twitter using logistic regression with particle swarm optimization. *Soft Comput.* **2021**, *25*, 11333–11345. [CrossRef]
33. Ajesh, F.; Aswathy, S.U.; Philip, F.M.; Jeyakrishnan, V. A Hybrid Method for Fake Profile Detection in Social Network Using Artificial Intelligence. In *Security Issues and Privacy Concerns in Industry 4.0 Applications*; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2021. [CrossRef]
34. Purba, K.R.; Asirvatham, D.; Murugesan, R.K. Classification of instagram fake users using supervised machine learning algorithms. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 2763–2772. [CrossRef]
35. Adewole, K.S.; Han, T.; Wu, W.; Song, H.; Sangaiah, A.K. Twitter spam account detection based on clustering and classification methods. *J. Supercomput.* **2018**, *76*, 4802–4837. [CrossRef]
36. Shalinda, A.; Kaushik, D. Identifying Fake Profiles in LinkedIn. *arXiv* **2022**, arXiv:2006.01381.
37. Wanda, P.; Jie, H.J. DeepProfile: Finding fake profile in online social network using dynamic CNN. *J. Inf. Secur. Appl.* **2020**, *52*, 102465. [CrossRef]
38. BalaAnand, M.; Karthikeyan, N.; Karthik, S.; Varatharajan, R.; Manogaran, G.; Sivaparthipan, C.B. An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter. *J. Supercomput.* **2019**, *75*, 6085–6105. [CrossRef]
39. Sahoo, S.R.; Gupta, B.B. Hybrid approach for detection of malicious profiles in Twitter. *Comput. Electr. Eng.* **2019**, *76*, 65–81. [CrossRef]
40. Aswani, R.; Kar, A.K.; Ilavarasan, P.V. Detection of Spammers in Twitter marketing: A Hybrid Approach Using Social Media Analytics and Bio Inspired Computing. *Inf. Syst. Front.* **2017**, *20*, 515–530. [CrossRef]
41. Kudugunta, S.; Ferrara, E. Deep neural networks for bot detection. *Inf. Sci.* **2018**, *467*, 312–322. [CrossRef]
42. Cao, J.; Fu, Q.; Li, Q.; Guo, D. Discovering hidden suspicious accounts in online social networks. *Inf. Sci.* **2017**, *394–395*, 123–140. [CrossRef]
43. Khan, S.A.; Iqbal, K.; Mohammad, N.; Akbar, R.; Ali, S.S.A.; Siddiqui, A.A. A Novel Fuzzy-Logic-Based Multi-Criteria Metric for Performance Evaluation of Spam Email Detection Algorithms. *Appl. Sci.* **2022**, *12*, 7043. [CrossRef]
44. Dewis, M.; Viana, T. Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails. *Appl. Syst. Innov.* **2022**, *5*, 73. [CrossRef]
45. Kaushik, K.; Bhardwaj, A.; Kumar, M.; Gupta, S.K.; Gupta, A. A novel machine learning-based framework for detecting fake Instagram profiles. *Concurr. Comput.* **2022**, *34*, e7349. [CrossRef]
46. Akyon, E.K.F.C. InstaFake Dataset. Available online: https://github.com/fcakyon/instafake-dataset (accessed on 25 February 2022).
47. Bakhshandeh, B. Instagram-Fake-Spammer-Genuine-Accounts. Available online: https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts (accessed on 14 March 2022).
48. Fan, D.; Sun, H.; Yao, J.; Zhang, K.; Yan, X.; Sun, Z. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy* **2020**, *220*, 119708. [CrossRef]
49. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
50. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
51. Ruder, S. An Overview of Gradient Descent Optimization Algorithms. 2016. Available online: http://arxiv.org/abs/1609.04747 (accessed on 9 October 2024).
52. Behera, R.K.; Jena, M.; Rath, S.K.; Misra, S. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Inf. Process. Manag.* **2020**, *58*, 102435. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.