Facebook Profile Credibility Detection using Machine and Deep Learning Techniques based on User's Sentiment Response on Status Message

Esraa A. Afify[1]*, Ahmed Sharaf Eldin[2]
Information Systems Department
Faculty of Computers and Artificial Intelligence
Helwan University, Cairo, Egypt

Ayman E. Khedr[3]
Information Systems Department
Faculty of Computers and Information Technology
Future University in Egypt (FUE), Cairo, Egypt

Abstract*—Recently, the impact of online Social Network sites (SNS) has dramatically changed, and fake accounts became a vital issue that has rapidly evolved. This issue gives rise to how to assess and measure the credibility of User-Generated Content (UGC). This content is used in finding trusted sources of information on SNS like Facebook, Twitter, etc. Consequently, classifying users' profiles and analyzing each user's behavior response based on the content generated became a challenge that must be solved. One of the most significant approaches is Sentiment Analysis (SA) which plays a major role in assessing and detecting the credibility degree of each user account behavior. In this paper, the aim of the study is to measure and predict the user's profile credibility by declaring the correlation degree among the UGC features that affect users' responses to status messages. The proposed models were implemented using six Supervised Machine Learning classifiers, an Unsupervised Machine Learning cluster model, and a Deep Learning Neural Network (NN) model. The research paper presents two experiments to evaluate Facebook profile credibility. At first, we applied a binary classification model to classify profiles into fake or genuine users. Then, we conducted a classification model on genuine users based on the credibility theory by using the Analytical Hierarchical Process (AHP) approach and computed the credibility score for each. Secondly, we selected and analyzed a public Facebook page (CNN public page) and obtained data from it for users' sentiment reactions and responses on statuses Messages relating to different topics on the period (2016/2017). Then, we performed LDA on the status corpus (Topic Modeling algorithm, Latent Dirichlet Allocation) to generate topic vectors. In addition, we performed Principal Component Analysis (PCA) method to visualize and classify each status topic distribution. Afterthought, we produced a status corpus cluster to classify users' behaviors through statuses posted and users' comments. As a conclusion of this study, the first experimental results achieved 95% and 99% accuracy to classify fake/genuine users and incredible/credible accounts, respectively. The second experiment outcome identified the clusters for the status corpus in 10 topic-features distribution and classified users' contents into credible or not according to the final calculated credibility score.

Keywords*—Fake profiles detection; credible profiles detection; sentiment analysis; supervised machine learning classifiers; unsupervised machine learning; binary classification; deep learning neural network; evaluation metrics

I. Introduction

Social Networks (SN) became the primary activity in our lives and turned out to be a virtual community [1]. In a real community, people massively exchange their opinions in every aspect of life. Some people could be considered as credible ones, and others are not according to the availability and reliability them. Usually, we accept other opinions according to the activeness behavior for each of them. Applying the same concept to the virtual SN community, people create posts and comments as if they are in real life through a variety of social accounts. Then, they interact with them in which raises the need to detect unreliable contents created in SNs [2].

Facebook and Twitter are Social Network Sites (SNS) that have experienced a dramatic increase in popularity over the last few years. Especially, Fake profiles on Facebook which harm privacy, online bullying, misuse, and trolling, etc. These profiles related to users with false credentials. It could be found through malicious and undesirable activities, causing problems for social network users. Users create fake profiles for social engineering, online representation to slander an individual, advertising, and campaigning for an individual or a group of individuals.

According to the Pew Research Center, Facebook has reached a leading position among the SNSs, with some worldwide active users amounting to over 2.3 billion as of July 2017. The main feature of Facebook and other SNSs is the possibility for users to share self-generated content like texts, pictures, audio, and video with their friends or followers. Users could create or share fake content because of missing approaches used to measure the credibility of the generated or shared content. On public pages of Facebook, users are not allowed to post, but they can only contribute by commenting on the posts. Sometimes users' input is unrelated to the post, for example, the topic of the post and the comment is different, or the comment is spam. Not only the comments of users on the page post are essential for measuring the credibility of the post, but also there are other features like the number of reactions, the number of shares, and Facebook emotions including "angry", "wow", "haha", "love", and "sad" reactions on posts, comments, and even messages,

*Corresponding Author

which could be used for measuring the credibility degree of the generated content.

The researchers found several characteristics and patterns that could be used to identify the credibility degree of user profile and user action/ interaction behavior. Then, they focused on, Sentiment Analysis (SA) which leads to figuring out how people feel about social media. With a sophisticated analysis of how people react to certain topics, we can predict various issues such as campaign success, marketing strategy, product messaging, customer service, and stock market price. As a result, we decided to take advantage of the recent extensions of reactions made by Facebook and do sentiment analysis on how people react differently to different posts. Based on the credibility theory, we used the Analytical Hierarchical Processes (AHP) approach to produce the feature weights to compute the credibility score for each user profile. After that, we analyzed users' sentiment analysis and performed LDA on the status corpus (Topic Modeling algorithm, Latent Dirichlet Allocation) to cluster topic-features distribution and Principal Component Analysis (PCA) method to visualize and classify each status topic distribution to compute a credibility score. Machine learning techniques contribute efficiently to detect semantic relations [3] in general and frauds [4] in specific. According to the revolution in Artificial Intelligence (AI), [5][6][7], we found that Machine Learning (ML) and Deep Learning (DL) are leading in research to predict the models' performance. For this reason in this research paper, we followed the ML and DL pipeline and performed two models for detecting the credible score of the users' profile and the content shared by them on social networks by discovering new patterns and characteristics for each user's profile. The first model is a binary classification model that automatically detects the fake and genuine profiles on Facebook. This model implies six supervised machine learning classifiers like Support Vector Machine, Random Forest, Decision Tree, K-Nearest Neighbor, Logistic Regression, Naïve Bayes, and a deep learning Neural Network model to classify the profiles into fake or genuine. The second model is a clustering model that detects credible and non-credible profiles according to user behaviors using the sentiment analysis generate on each profile. This model applied using the K-Means unsupervised machine learning clustering. Different performance analysis approaches conducted to evaluate both experiments such as plotting the Learning Curves (LC), calculating the "Area Under the Curve" (AUC) of "Receiver Characteristic Operator" (ROC), illustrating the ROC/AUC Curves, computing the Confusion Metrics (CM), and generating classification reports to summarize results for each applied classifier.

Research paper organization. This paper is organized as follows: Section II briefly discusses the related works to the research study. Section III presents the research methodologies. Section IV describes the proposed methodology. Section V identifies the results and discussion of the experiments. And Section VI provides the research study conclusion.

II. Literature Review

Extracting semantic relations has been successfully applied. As found in, Sultan et al. (2012) [8], semantic relations exchange is performed for information sources' collaboration. This approach would support different sources including Facebook for detection. Another research in a different direction, as in Sharaf Eldin et al. (2015) [9], focused on detecting the appropriate technique for the type of data as successful techniques determination is one of the key success factors.

Focusing on Facebook sources concerning credibility detection on Facebook, the most recent researches are: Lê et al. (2019) [10], proposed a ranking scheme for fake Facebook user accounts detection. The model includes both feature-based approaches and graph-based approaches by utilizing the SVM and SybilWalk algorithm.

Smruthi et al. (2019) [11], used a hybrid model based on machine learning and skin detection algorithms to detect the existence of fake accounts on Facebook. The model result achieved 80% accuracy by utilizing the supervised machine learning algorithms.

Gupta et al. (2017) [1], attempted to detect fake accounts on Facebook based on user profile activities and interactions. The model result achieved 79% accuracy by applying the most supervised machine learning algorithms.

Wani et al., (2016) [12], presented a novel approach to predict fake profiles on Facebook. The model was trained using supervised machine learning algorithms. The theoretical machine learning model has been proposed to classify the user profiles into fake and genuine.

Saikaew et al., (2015) [2], developed a system for measuring credibility on Facebook information. At first, the authors proposed a FB credibility evaluator. Secondly, they developed a chrome extension to evaluate the credibility of each post. Based on the usage analysis of their FB credibility chrome extension, about 81% of users responded agree with suggested credibility automatically computed by the proposed system.

III. Research Methodologies

A. Machine Learning (ML): Overview

ML is the main branch of computer science that, provides computers with the capacity to learn without being programmed. It begins with data extracting knowledge. In *ML*, a dataset of observations, called vectors, comprises several variables called features or attributes [13].

In the next sections, we will discuss the two main categories of machine learning, which are supervised learning and unsupervised learning. In this paper, we used supervised learning for the first experiment and unsupervised learning for the second one.

B. Supervised Learning: Methods

Supervised learning, also known as predictive modeling, is the process of making predictions using pre-labeled data. As shown in Fig. 1, it takes input datasets with output labels. This data called 'training data' that include a set of training examples [14]. A subclass of supervised learning problems is binary classification, where there are only two labels for class features as a fake class or genuine class.

In our first proposed model, the dataset is a series of fake and genuine users' profiles, our supervised task is to predict whether each user account is fake or genuine. First, we train a classifier using the existing label. Labeled data with the desired output is called 'model training' because the model is learning the relationship between the attributes (features) of the data and the desired output value (target). These features include the number of friends, number of followers, statuses, gender, and language, and so on. Second, we make predictions for the new data for which we do not know the true outcome. For example, when a new user account created, we want our trained model to accurately predict whether the user account is fake or genuine without a human examination. The best-case scenario will allow the classifier to correctly set the class labels for unseen cases. This is supervised learning because there is a specific outcome we are trying to predict, in our work namely, fake, or genuine users.

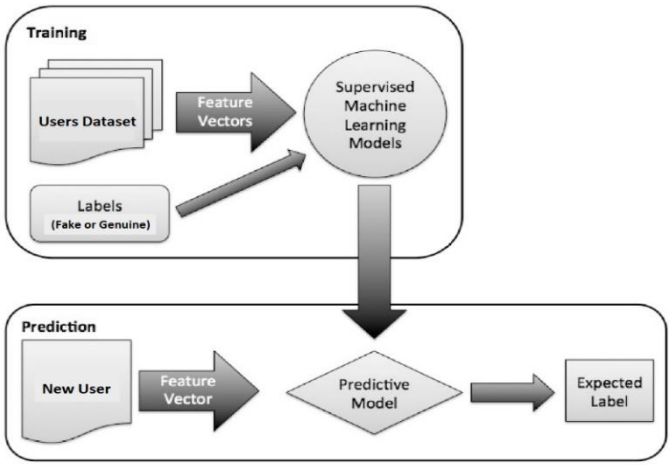

Fig. 1. Supervised Machine Learning.

In the next section, we will discuss briefly the six classification models that have been selected to implement the research work on this paper.

C. Classification Models: Brief

Classification, known as an instance of Supervised Machine Learning, is a method of setting to which class does a new observation belongs, based on training the machine with an existing data containing observations, in which class is predefined. The algorithms which implement Classification are called as Classifiers, there are many types of classifiers available, as follows:

1) Decision Tree (DT) Classifier applies a hierarchical structure, each internal node denotes a test on an attribute. It breaks down the dataset to build the model. Each node classifies an output value of a test and every leaf or terminal node holds a class label. This classifier splits the tree in the target variable that is most dominant, after calculating the entropy and gain scores [15] [16].

$$\text{Entropy (s)} = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (1)$$

$$\text{Gain (S, A)} = \underbrace{\text{Entropy(S)}}_{\text{Original entropy of S}} - \underbrace{\sum_{v \in values(A)} \frac{|Sv|}{|S|} . Entropy\,(Sv)}_{\text{relative entropy of S}} \quad (2)$$

2) Random Forest (RF) Classifier based on ensemble learning. It combines multiple decision trees to form a strong classifier [17]. In each decision tree, we pick a random sample from the training set, then choose random features at each node of the tree. After that, we split the tree at the best split among the selected features. In the binary classification case, the result is the percentage of trees that give a majority voting score.

$$RFfi_i = \frac{\sum_j norm\ fi_{ij}}{\sum_{j \in all\ features, k \in all\ trees} normfi_{jk}} \quad (3)$$

3) Logistic Regression (LR) Classifier uses a *sigmoid function* [7] [15], as shown in *Fig. 2.* It maps predicted observations to estimate probabilities between 0 and 1 or True/False.

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \quad (4)$$

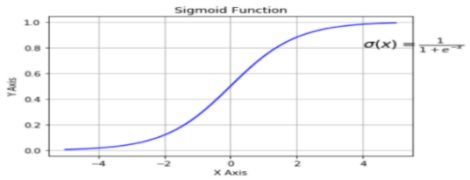

Fig. 2. Sigmoid Function.

4) K-Nearest Neighbor (KNN) Classifier based on similarity measures or distance functions. It uses a *K* value to get the nearest neighbor class, then performs a majority voting. The KNN calculates the numerical values using *distance formulas* [7] [18] [19].

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{x} (x_t - y_t)^2} \quad (5)$$

$$\text{Manhattan distance} = \sum_{i=1}^{k} |x_i - y_i| \quad (6)$$

$$\text{Minkowski distance} = \left(\sum_{i=1}^{k} (|x_i - y_i|)^q\right)^{1/q} \quad (7)$$

5) Naïve Bayes (NB) Classifier based on Bayes' Theorem and conditional probability. It uses Bayes' formula to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction [7] [15].

$$p(c|x) = \frac{p(x|c)\, p(c)}{p(x)} \quad (8)$$

$$p(c|x) = p(x_1|c) \times p(x_2|c) \times \ldots \times p(x_n|c) \times p(c) \quad (9)$$

- P $(c|x)$ is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P $(x|c)$ is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

6) Support Vector Machine (SVM) Classifier plots each observation as a point in n-dimensional space (n refers to features). After that, it finds the optimal hyper-plane by maximizing the margins between classes, as shown in Fig. 3, [20][21].

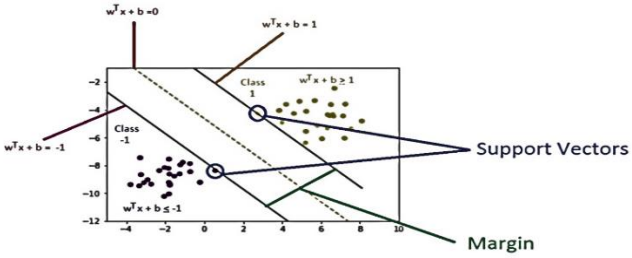

Fig. 3. Support Vector Machine.

D. Artificial Neural Network Deep Learning: Overview

Deep learning, known as, a subset of machine learning that does a similar function, but there are many layers, every layer provides a different performance to the data it feeds on, as shown in Fig. 4, for example. The name Artificial Neural Network (ANN) came from, functioning as an inspiration, or as it works as the function of the neural networks present in the human brain [5] [6] [22]. Recently, deep learning is the evolution of machine learning, which performs as a neural network that vest machines to produce accurate decisions without humans interfering.

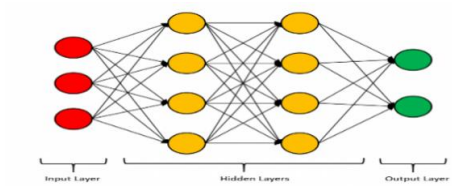

Fig. 4. ANN with Two Hidden Layers.

E. Unsupervised Learning: Methods

Unsupervised learning, also known as a data-driven model, is the process of identifying clusters using unlabeled data. It takes input dataset only where patterns or structures are found as hidden features among the dataset. This training dataset is a collection of observation examples without a specific desired outcome. Clustering is a typical example of unsupervised learning that finds visual classifications that match hypotheses. The purpose of clustering is to bring similarities, regardless of the data class. Therefore, a clustering algorithm usually, needs to know how to calculate the similarity, then start to run.

K-Means Clustering is a clustering algorithm that combines the n of observations into k clusters that aggregated with each other, according to specific similarities [6] [23], as shown in Fig. 5.

It works according to three steps, as follows:

- Initialization – K initial "means" (centroids) generated randomly.
- Assignment – K clusters created by associating each observation with the nearest centroid.
- Update – The centroid of the clusters becomes the new mean.

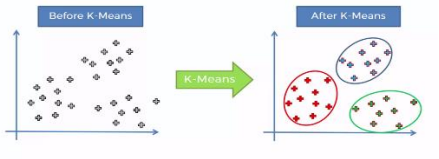

Fig. 5. K-Means Clustering.

F. Evaluation Curves and Metrics for a Classification Model

We can evaluate the classification model with different curves and metrics, such as Learning Curves, AUC-ROC Curves, Confusion Matrix, Accuracy Score, Precision Score, Recall Score, F1 Score, and Specificity [24].

Learning Curve: used to plot each classification model. These plots used for visualizing the observations with the metric performance. Line of learning plotted the y-axis over the experience of the x-axis to model the training set performance against the set as a function of the training set size. In a learning curve, a good fit is clarified by a training and validation loss that decreased to a point of stability with a small gap between both final loss outputs.

ROC/AUC Curve: used to plot the 'true positive rate' illustrated on the y-axis versus the 'false positive rate' which illustrated on the x-axis for the whole potential classification thresholds.

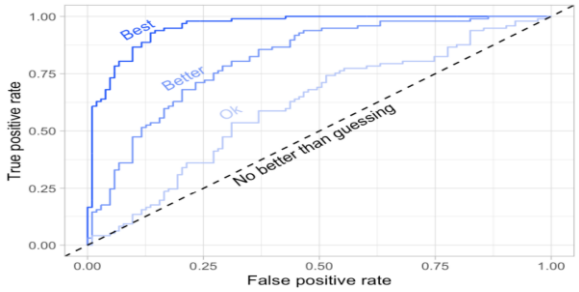

Fig. 6. ROC / AUC Curve.

As shown in Fig. 6, to utilize this terminology, 'sensitivity' defined on the y-axis and 1 minus specificity on the x-axis for every classification threshold from zero to one. Also, the dashed line in the graph is the baseline state the random guesses where the 'true positive rate' increases linearly with the 'false positive rate', and its AUC is 0.5; the blue line is the ROC plot of the model, and its AUC is less than 1. In a perfect case, the 'true positive' samples have a probability 1, so that

the ROC starts at the point with 100% 'true positive' and 0 'false positives'. The AUC of such a perfect curve is 1. A line that is diagonal from the lower left corner to the upper right corner represents a random guess. The higher the line is in the upper left-hand corner, the better.

Confusion Matrix: is a table with four different combinations of predicted and actual values. It illustrates all the observations in the testing set. In other words, it summarizes predicted outcomes and true outcomes for testing, as presented in Table I.

TABLE I. CONFUSION MATRIX

		Predicted		
		Negative	Positive	TN = True Negative FP = False Positive FN = False Negative TP = True Positive
Actual	Negative	TN	FP	
	Positive	FN	TP	

- TN is the false sample, which is predicted to be false by the model.
- FP is the false sample, which is predicted to be true by the model.
- FN is the positive sample, which is predicted to be false by the model.
- TP is the positive sample, which is predicted to be true by the model.

The calculation formulas of *FPR* and *TPR* are as follows:

$$FPR = \frac{FP}{TN+FP}, TPR = \frac{TP}{TP+FN} \quad (10)$$

$$Accuracy = \frac{|TP|+|TN|}{|TP|+|TN|+|FP|+|FN|} \quad (11)$$

$$Precision = \frac{|TP|}{|TP|+|FP|} \quad (12)$$

$$Recall = \frac{|TP|}{|TP|+|FN|} \quad (13)$$

$$F1 = 2.\frac{Precision.Recall}{Precision+Recall} \quad (14)$$

$$Specifity = \frac{|TN|}{|TN|+|FP|} \quad (15)$$

$$AUC = \frac{\sum(n_0+n_1+1-r_i)-n_0(n_0+1)/2}{n_0 n_1} \quad (16)$$

IV. PROPOSED METHODOLOGY

In this research paper, we aim to propose a set of minimum features that can detect credible users' profiles and sentiment responses with the highest accuracy into two models. To do that, we followed the general machine learning and deep learning pipeline step-by-step, as shown in Fig. 7.

A. Data Acquisition: Datasets

Two different datasets had been used to implement our proposed models. Firstly, we applied the classification model on a public dataset that consists of 2818 fake and genuine users' profiles with 34 features, but after applying the correlation for them, we extracted 7 features that affect the detection method. Secondly, the cluster model had been applied on a CNN public Facebook page. This data related to various users' sentiment responses at 10 different topics distribution on status messages during the period (2016/2017). The dataset consists of 9282 status messages, with 14 features. The experiments implementation was deployed by python code on Google Colab Notebooks and applied using Machine Learning models with the help of the Scikit-learn libraries. Keras with TensorFlow used for Deep Learning model.

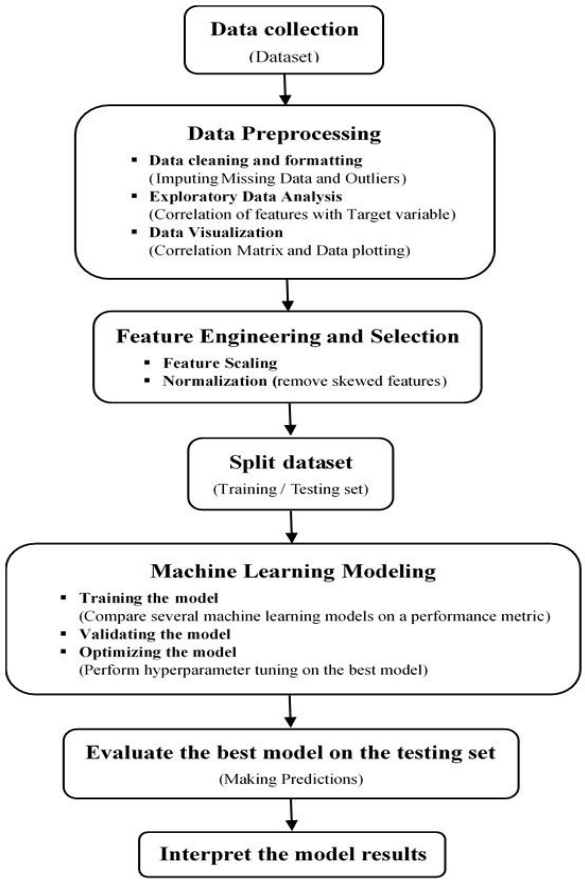

Fig. 7. ML and DL Pipeline.

B. Data Pre-Processing

1) Data cleaning using outliers detection: The Tukey's boxplot method [25], as shown in Fig. 8, considered to be one of the most frequently used methods for finding outliers uses the interquartile range with boxplot to filter out exceptionally large or ridiculously small numbers whether a distribution is skewed and whether there are potential unusual observations in the dataset.

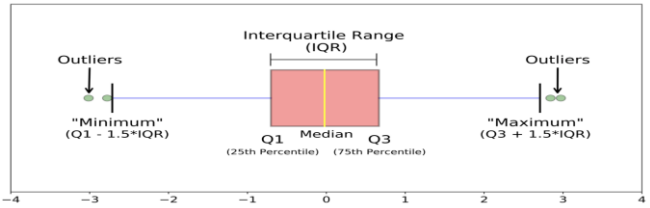

Fig. 8. Tukey's Method (Box Whisker).

Algorithm 1: Pseudo-code for Outliers Detection using the Tukey Method

Input: Dataset

Output: Suspected Outliers data points

Procedure: First quartile 25% (Q1), Third quartile 75% (Q3), Interquartile Range 50% (IQR)

1: for data values d_i in the Training dataset **do**
2: Arrange $d_i \rightarrow$ Q1 and $d_i \rightarrow$ Q3
3: Compute IQR = Q3 – Q1
4: **Compute the outlier boundaries formulas, as follows:**
5: Lower Outlier Boundary l_i = Q1 – 1.5 (IQR)
6: Upper Outlier Boundary u_i = Q3 + 1.5 (IQR)
7: **if** $d_i < l_i$ or $d_i > u_i$ **then**
8: **return** Outliers
9: **end if**
10: end for

We have detected and eliminated the outliers from the Facebook CNN public page dataset following Algorithm 1 proved above. We visualized the boxplots and removed all outliers in each user sentiment response to achieve the best results during the experiment testing, as shown in Fig. 9 and 10, respectively.

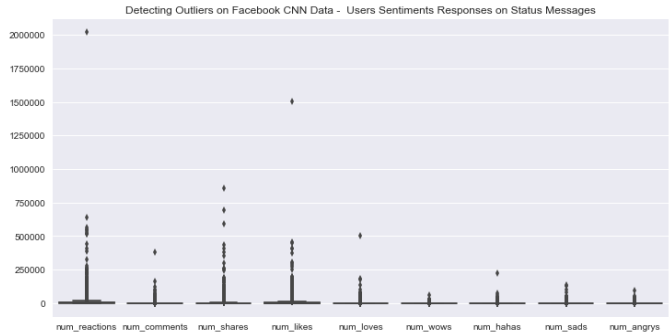

Fig. 9. Detecting Outliers on Facebook CNN Page - Users' Responses.

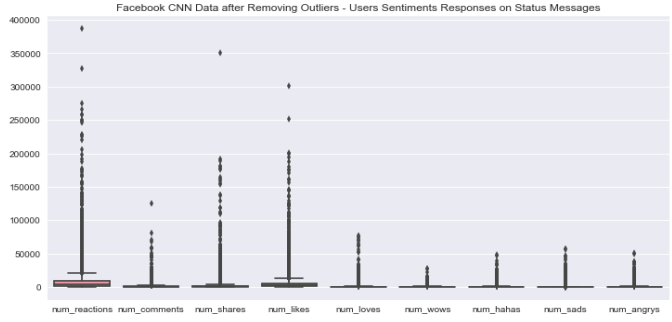

Fig. 10. Eliminating Outliers from Facebook CNN Data - Users' Responses.

2) Data analysis: elbow method and silhouette score method: A fundamental step for any unsupervised algorithm is to determine the optimal k number of clusters into which the data may be clustered. The Elbow Method is one of the most popular methods used to determine this optimal value of k, as shown in Algorithm 2 and Fig. 11.

Algorithm 2: Pseudo-code for Elbow & Silhouette Method

Input: Data $X = \{\mathbf{X_1}, \ldots, \mathbf{X_n}\}$, the order k, **MAX** number of allowed iterations

Output: A partition $P = \{C_1, \ldots, C_k\}$

1: $t = 0$, $\mathbf{P} = \emptyset$
2: Randomly initialize μ_i, $i = 1, \ldots, k$
3: **loop**
4: $t + = 1$
5: Assignment Step: assign each sample $\mathbf{x}_j$ to the cluster with the nearest representative
6: $c_i^{(t)} = \{\mathbf{X_j} : d(\mathbf{X_j}, \mu_i) \leq d(\mathbf{X_j}, \mu_h)$ for all $h = 1, \ldots, k\}$
7: Update Step: update the representatives
8: $\mu_i^{(t+1)} = \frac{1}{|c_1^{(t)}|} \sum_{xj} \in c_i^{xj}$
9: Update the partition with the modified clusters:
$P^t = \{c_1^{(t)}, \ldots, c_k^{(t)}\}$
10: **if** $t \geq$ **MAX** OR $P^t = P^{t-1}$ **then**
11: **return** P^t
12: **end if**
13: end loop

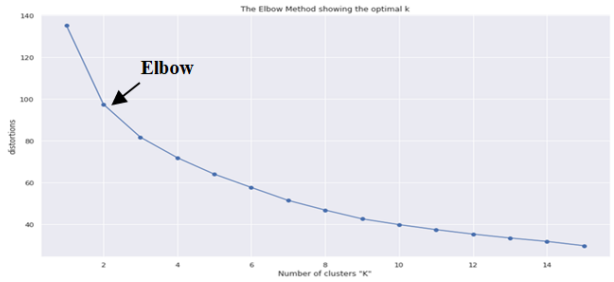

Fig. 11. Selecting the Number of Clusters k using the "Elbow Method".

In this paper the researchers had used the 'elbow method' to specify the number of clusters k that the algorithm must find to define the user's profiles groups numbers. This curve has the shape of an arm, the "elbow" found at k=2 in this model. Where the distortions illustrated on the y-axis, then dropped very quickly as the k increased up to 2, then it decreased much more slowly as the k increased more, which illustrated on the x-axis.

In Fig. 12, the observations divided into two groups of users:

- ***Credible users:*** *'cluster 0'*, this group of users are not extensively using Facebook a lot and only use it for surfing. The reaction count is only 48 on posts and comments 3. And they did not share any posts and only react 47 '*like*' on posts.
- ***Non-credible users:*** *'cluster 1'*, this group of users are extensively using Facebook. They react to 82067 posts and comments 57770. And they share posts and use the other reacts on posts.

The Silhouette considered being a better method to choose the optimal number of clusters k to be formulated from the data. This method measures the similarity of a data instance within a cluster comparing with another cluster. Then computes the score for each data instance and calculate the formula for the Silhouette coefficient as shown in Fig. 13.

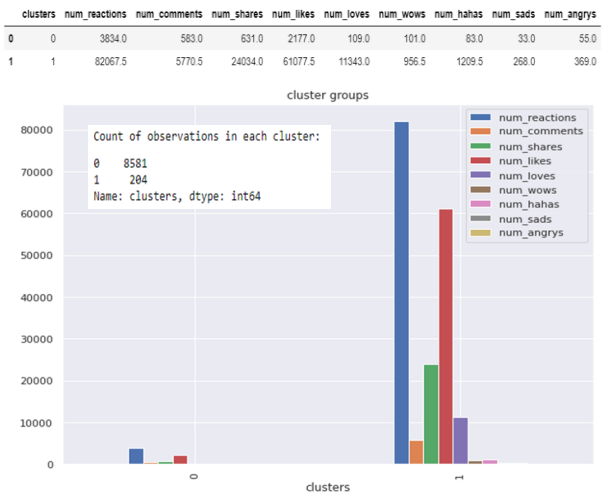

Fig. 12. Visualizing the Clusters Groups.

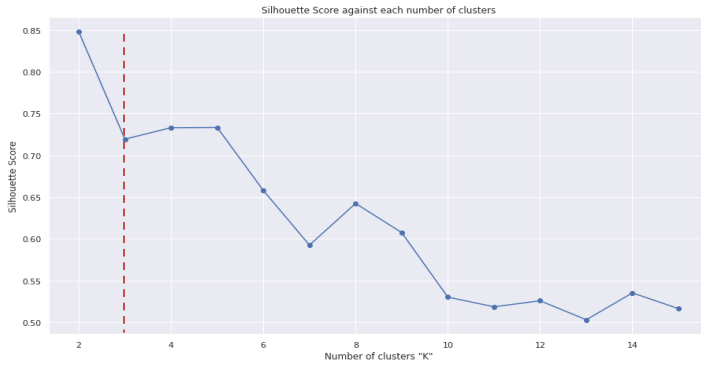

Fig. 13. Selecting the Number of Clusters k using the "Silhouette Score".

From the pivoted data frame shown in Fig. 14, we can see that there are three groups of Facebook users:

- ***Group 0:*** This is the group of people who, according to the provided dataset, happen to use Facebook quite a lot. But they are the kind of people who usually give people the 'like' react mostly.
- ***Group 1:*** Which indicates that the user of this group might not use Facebook a lot or use it only for surfing. Their number of reactions are around 3375 and comments only 511. They do not share a lot of posts. And mostly they use 'like' react on posts.
- ***Group 2:*** This group also shows that people use Facebook a lot. These people tend to comment and share the posts a lot. They also tend to use other reacts on posts besides the 'like' react.

3) Data visualization: correlation coefficient matrix: Visualizing the Correlation Matrices after dealing with null values, dropping unnecessary features, and eliminating outliers from the dataset, as shown in Fig. 15 and 16.

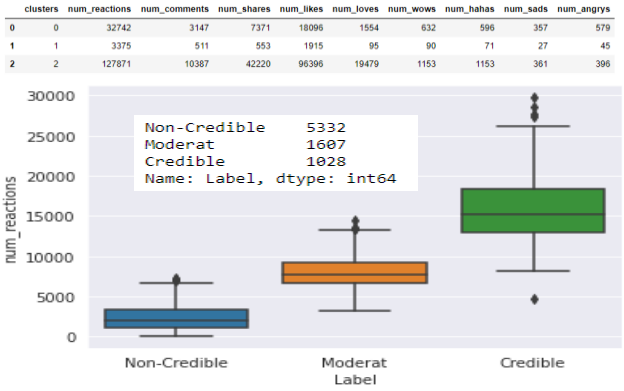

Fig. 14. Visualizing the Clusters Groups from the Pivoted Data Frame.

Model 1

	statuses_count	followers_count	friends_count	favourites_count	listed_count	sex_code	lang_code
statuses_count	1.000000	0.046942	0.368709	0.489355	0.259307	0.041663	0.232903
followers_count	0.046942	1.000000	0.077779	0.025199	0.650564	0.005834	0.039415
friends_count	0.368709	0.077779	1.000000	0.276687	0.311310	0.043719	-0.002270
favourites_count	0.489355	0.025199	0.276687	1.000000	0.078469	0.015427	0.153274
listed_count	0.259307	0.650564	0.311310	0.078469	1.000000	0.028808	0.043786
sex_code	0.041663	0.005834	0.043719	0.015427	0.028808	1.000000	0.159291
lang_code	0.232903	0.039415	-0.002270	0.153274	0.043786	0.159291	1.000000

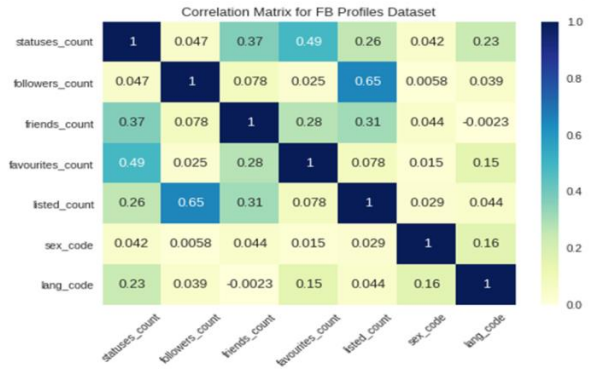

Fig. 15. Facebook user Profile Correlation Matrix.

Model 2:

	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys
num_reactions	1.000000	0.565057	0.761640	0.970389	0.847115	0.402841	0.368019	0.311652	0.266160
num_comments	0.565057	1.000000	0.518836	0.467801	0.454583	0.305204	0.381597	0.139033	0.497097
num_shares	0.761640	0.518836	1.000000	0.718402	0.571698	0.443319	0.369357	0.259377	0.264474
num_likes	0.970389	0.467801	0.718402	1.000000	0.849888	0.338090	0.282883	0.182000	0.094659
num_loves	0.847115	0.454583	0.571698	0.849888	1.000000	0.187003	0.148161	0.100561	0.047265
num_wows	0.402841	0.305204	0.443319	0.338090	0.187003	1.000000	0.169756	0.093587	0.263709
num_hahas	0.368019	0.381597	0.369357	0.282883	0.148161	0.169756	1.000000	-0.019194	0.144948
num_sads	0.311652	0.139033	0.259377	0.182000	0.100561	0.093587	-0.019194	1.000000	0.217023
num_angrys	0.266160	0.497097	0.264474	0.094659	0.047265	0.263709	0.144948	0.217023	1.000000

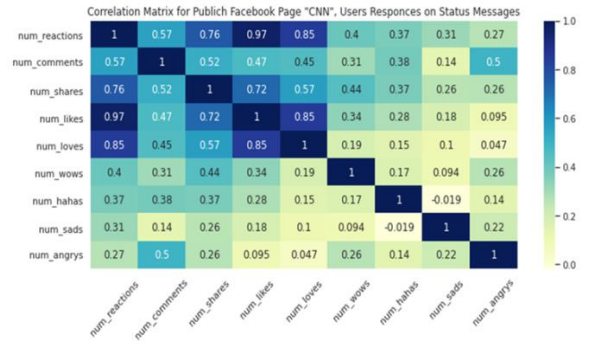

Fig. 16. Facebook CNN Page - Users' Sentiments Correlation Matrix.

C. Feature Engineering and Selection: Steps

Each feature in the dataset has a degree of importance to represent the data very well. In consequence, the feature selection step is needed like a filter, wrapper, and embedded method. One of the Topic Modeling methods is Latent Dirichlet Allocation (LDA). LDA is used to classify text in a collection of group documents by topics, as described in the following steps and shown in Algorithm 3.

The step-by-step approach for LDA with classifiers explained below:

- Read the data which comprises a combination of genuine and fake users.
- Pre-process the data to filter out status messages in genuine users' case.
- Prepare every user data by concatenating entire posts for user.
- Apply the LDA algorithm on posts after concatenation to generate topics.
- Generates user or post probabilities of *n* topics.
- Evaluate the loss and Goss metrics for every user or post.
- Use the vectors set of features for training classifiers.
- Classify the feature vector into train/test set then train with models.
- Report and compute accuracy, recall, f-score and precision of the algorithm.

Algorithm 3: Pseudo-code for LDA

1: Choose distribution of topic

2: θa ~Dirichlet (α) where $a \in \{1, \ldots., X\}$ and Dirichlet (α) is the Dirichlet distribution for α parameter

3: For every word W_{ab} in the document where $b \in \{1, \ldots ... N_a\}$

4: Select a particular topic $z_{ab} \sim Multi\ (\theta a)$ where multi ( ) is a multinomial

5: Select a word $W_{ab} \sim \beta Z_{ab}$

where w indicates words, Z indicates topic vector and β is a K x V matrix of word probability for every term (column) and every topic (row) and $\beta_{ab} = P(W_a = 1|Z^a=1)$

In addition, dimensionality reduction considered a type of feature selection applied for significant large features. One of the most well-known dimensionality reduction methods called Principal component analysis (PCA). PCA method is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [26]. PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to visualize genetic distance and relatedness between populations. PCA is either done in the following five steps as shown in Algorithm 4.

Algorithm 4: Pseudo-code for PCA

1: Compute the mean feature vector

$\mu = \frac{1}{p} \sum_{k=1}^{p} x_k$, where, x_k is a pattern

2: Find the covariance matrix

$C = \frac{1}{p} \sum_{k=1}^{p} \{x_k - \mu\}^T$ where, T represents matrix transposition

3: Compute Eigen values λ_i and Eigen vectors v_i of covariance matrix

$Cv_i = \lambda_i v_i$ *(i = 1, 2, 3, ...q),* q = number of features

4: Estimating high-valued Eigen vectors

(i) Arrange all the Eigen values (λ_i) in descending order

(ii) Choose a threshold value, θ

(iii) Number of high-valued λ_i can be chosen to satisfy the relationship

$[\sum_{i=1}^{s} \lambda_i]\ [\sum_{i=1}^{p} \lambda_i]^{-1} \geq \theta$, where, s = number of high valued λ_i chosen

(iv) Select Eigen vectors corresponding to selected high valued λ_i

5: Extract low dimensional feature vectors (principal components) from raw feature matrix.

$P - V^T x$, where, V is the matrix of principal components and x is the feature matrix

The first proposed model consists of various steps:

- Determines the main account features that influence a correct detection of fake profiles,
- Apply and compare different classification algorithm,
- Illustrate and compute the evaluation curves and metrics for each classifier, and.
- Compute the credibility score for the genuine users' accounts by using the AHP approach, as shown in algorithm 5.

The second model also consists of various steps:

- Select and acquire data from a public Facebook page for user sentiment analysis,
- Determine the main features that influence users' profile behaviors, through status message and users' responses,
- Perform LDA topic modeling algorithm on status corpus and generate topic vectors,
- Assign for each status a most relevant topic label based on highest probability,
- Perform PCA to visualize topic distribution and correlation matrix,
- Analyze and visualize users' responses on each topic,
- Apply a K-Mean clustering algorithm to cluster status corpus using topic-features,
- Plot likelihood/inertia for each K-number of clusters for each method, and

- Compute the credibility score for users' responses on status corpus by using the AHP approach.

Algorithm 5: Pseudo-code for AHP Approach

Input: Dataset

Output: Alternatives Ranking

Procedure:

1: for data values d_i in the Training dataset **do**

2: **- Construct the AHP Hierarchy for evaluation:**
Level 1 → define a decision goal
Level 2 → set the criterion
Level 3 → distribute the alterative

3: **- Calculate the Pairwise Comparison Matrix (Matrix A)**

$$A=\begin{bmatrix} a11 & a12 & \cdots & a1n \\ a21 & a22 & \cdots & a2n \\ \vdots & \vdots & \cdots & \vdots \\ an1 & an2 & \cdots & ann \end{bmatrix}, \text{ where: } a_{ij} = 1/a_{ji} \text{ (I,j = 1,2,..,n)}$$

4: **- Calculate Normalized principal Eigen Vector of Matrix A 'w' (Priority Vector Matrix**

$$e^T = (1, 1, \ldots..,1) \rightarrow W = \lim_{k\to\infty} \frac{A^K .e}{e^T .A^K .e}$$

$$Aw = \lambda_{ajwj-n} w \rightarrow \lambda max \geq n$$

$$\lambda max = \frac{\sum}{w1}$$

A={aij} with aij=1/ aij

Where:

A → pair wise comparison
W → normalized weight vector
λmax → maximum eigen value of matrix A
aij → numerical comparison between the values i and j

5: **- Calculate the weights and testing the consistency for each level**

6: **- Calculate Consistency Ratio**

$$Calculate\ Consistency\ Ratio\ (CR) = \frac{Consistency\ Index\ (CI)}{Random\ Consistency\ Index\ (RI)} \rightarrow CR = \frac{CI}{RI},$$

Where:

$$CI = \frac{\lambda max - n}{n-1}$$

7: **if Matrix Consistence, CR ≤ 0.10 then**
Get the priorities of all selection criteria
Get the rank of each alternative with respect to the selection criteria
8: **return Get the overall rank of the alternatives**
9: **end if**
10: end for

D. Credibility Detection Method: Formulas

In the proposed model, we proposed a credibility formula for both genuine profiles and status messages. This formula contains several parameters each of these parameters multiplied with a specific weight define according to the correlation coefficient matrix. These weights computed according to the Analytical Hierarchical Process (AHP) approach, which depends on the credibility theory. Applying this equation will lead us to rank users' accounts each according to the credibility ranking. Consequently, we can predict the degree of trust and credibility of Facebook user profiles, as shown in Fig. 17 and 18.

1) Facebook Profile Credibility Formula:

Profile Credibility Degree = Statues count * 0.33
+ Followers count * 0.23
+ Friends count * 0.16
+ Favorites count * 0.13
+ List count * 0.08
+ Gender code * 0.04
+ Language code * 0.02 (17)

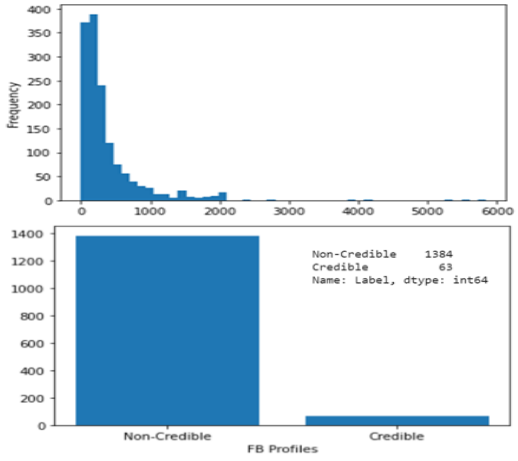

Fig. 17. Count of Credible users' Profiles Plot.

2) Facebook Status Message Credibility Formula:

Status Credibility Degree = $num_{reactions}$ * 0.26
+ $num_{comments}$ * 0.189
+ num_{shares} * 0.12
+ num_{likes} * 0.17
+ num_{loves} * 0.107
+ num_{wows} * 0.075
+ num_{hahas} * 0.046
+ num_{sads} * 0.027
+ num_{angrys} * 0.014 (18)

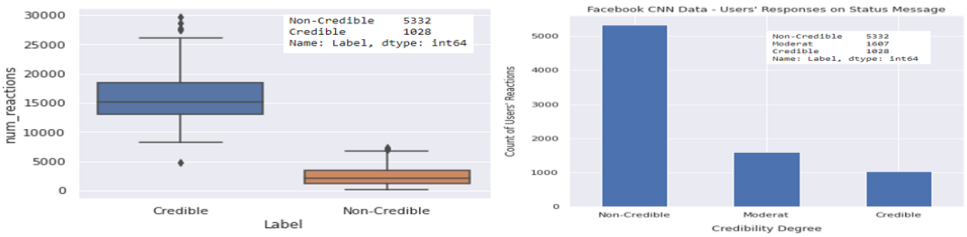

Fig. 18. Count of Credible users' Responses Plot.

E. AHP: Calculation

In the following Fig. 19 and 20, we will present how weights computed according to the Analytical Hierarchical Process (AHP) approach in details.

Pairwise comparisons

	statuses_count	followers_count	friends_count	favourites_count	listed_count	sex_code	lang_code
statuses_count	1.00	7.00	4.00	3.00	5.00	9.00	5.00
followers_count	0.14	1.00	7.00	7.00	2.00	8.00	7.00
friends_count	0.25	0.14	1.00	5.00	4.00	8.00	9.00
favourites_count	0.33	0.14	0.20	1.00	7.00	7.00	6.00
listed_count	0.20	0.50	0.25	0.14	1.00	7.00	7.00
sex_code	0.11	0.13	0.13	0.14	0.14	1.00	6.00
lang_code	0.20	0.14	0.11	0.17	0.14	0.17	1.00
Sum	2.24	9.05	12.69	16.45	19.29	40.17	41.00

STANDARDIZED MATRIX

	statuses_count	followers_count	friends_count	favourites_count	listed_count	sex_code	lang_code	**Weight**
statuses_count	0.45	0.77	0.32	0.18	0.26	0.22	0.12	33.2%
followers_count	0.06	0.11	0.55	0.43	0.10	0.20	0.17	23.2%
friends_count	0.11	0.02	0.08	0.30	0.21	0.20	0.22	16.2%
favourites_count	0.15	0.02	0.02	0.06	0.36	0.17	0.15	13.2%
listed_count	0.09	0.06	0.02	0.01	0.05	0.17	0.17	8.1%
sex_code	0.05	0.01	0.01	0.01	0.01	0.02	0.15	3.7%
lang_code	0.09	0.02	0.01	0.01	0.01	0.00	0.02	2.3%

CI and CR worksheet

	statuses_count	followers_count	friends_count	favourites_count	listed_count	sex_code	lang_code	SUM	SUM/Weight
statuses_count	0.33	1.63	0.65	0.40	0.41	0.34	0.11	3.86	11.63
followers_count	0.05	0.23	1.14	0.92	0.16	0.30	0.16	2.96	12.76
friends_count	0.08	0.03	0.16	0.66	0.33	0.30	0.21	1.77	10.89
favourites_count	0.11	0.03	0.03	0.13	0.57	0.26	0.14	1.28	9.66
listed_count	0.07	0.12	0.04	0.02	0.08	0.26	0.16	0.74	9.14
sex_code	0.04	0.03	0.02	0.02	0.01	0.04	0.14	0.29	7.82
lang_code	0.07	0.03	0.02	0.02	0.01	0.01	0.02	0.18	7.89

Saaty's CIr values for matrices are given by the following table

Size of Matrix	Random Consistency (CIr)	
1	0	1
2	0	2
3	0.58	3
4	0.90	4
5	1.12	5
6	1.24	6
7	1.32	7
8	1.41	8
9	1.45	9
10	1.49	10

count	7.00
lambda max	9.968
CI	0.495
CR	**0.37**
constant	1.32

Fig. 19. Facebook Profile Credibility Weights (Model 1).

Pairwise comparisons

	num_reactions	num_comment	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys
num_reactions	1.00	7.00	4.00	3.00	3.00	4.00	7.00	8.00	8.00
num_comments	0.14	1.00	6.00	8.00	2.00	8.00	3.00	7.00	7.00
num_shares	0.25	0.17	1.00	8.00	0.50	3.00	4.00	8.00	3.00
num_likes	0.33	0.13	0.13	1.00	8.00	8.00	8.00	9.00	9.00
num_loves	0.33	0.50	2.00	0.13	1.00	5.00	5.00	8.00	7.00
num_wows	0.25	0.13	0.33	0.13	0.20	1.00	7.00	9.00	8.00
num_hahas	0.14	0.33	0.25	0.13	0.20	0.14	1.00	8.00	7.00
num_sads	0.13	0.14	0.13	0.11	0.13	0.11	0.13	1.00	8.00
num_angrys	0.13	0.14	0.33	0.11	0.14	0.13	0.14	0.13	1.00
Sum	2.70	9.54	14.17	20.60	15.17	29.38	35.27	58.13	58.00

STANDARDIZED MATRIX

	num_reactions	num_comment	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	**Weight**
num_reactions	0.37	0.73	0.28	0.15	0.20	0.14	0.20	0.14	0.14	26.0%
num_comments	0.05	0.10	0.42	0.39	0.13	0.27	0.09	0.12	0.12	18.9%
num_shares	0.09	0.02	0.07	0.39	0.03	0.10	0.11	0.14	0.05	11.2%
num_likes	0.12	0.01	0.01	0.05	0.53	0.27	0.23	0.15	0.16	17.0%
num_loves	0.12	0.05	0.14	0.01	0.07	0.17	0.14	0.14	0.12	10.7%
num_wows	0.09	0.01	0.02	0.01	0.01	0.03	0.20	0.15	0.14	7.5%
num_hahas	0.05	0.03	0.02	0.01	0.01	0.00	0.03	0.14	0.12	4.6%
num_sads	0.05	0.01	0.01	0.01	0.01	0.00	0.00	0.02	0.14	2.7%
num_angrys	0.05	0.01	0.02	0.01	0.01	0.00	0.00	0.00	0.02	1.4%

CI and CR worksheet

	num_reactions	num_comment	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	SUM	SUM/Weight
num_reactions	0.26	1.32	0.45	0.51	0.32	0.30	0.32	0.22	0.11	3.81	14.67
num_comments	0.04	0.19	0.67	1.36	0.21	0.60	0.14	0.19	0.10	3.50	18.52
num_shares	0.07	0.03	0.11	1.36	0.05	0.22	0.19	0.22	0.04	2.29	20.50
num_likes	0.09	0.02	0.01	0.17	0.85	0.60	0.37	0.25	0.13	2.49	14.64
num_loves	0.09	0.09	0.22	0.02	0.11	0.37	0.23	0.22	0.10	1.46	13.66
num_wows	0.07	0.02	0.04	0.02	0.02	0.07	0.32	0.25	0.11	0.93	12.38
num_hahas	0.04	0.06	0.03	0.02	0.02	0.01	0.05	0.22	0.10	0.55	11.79
num_sads	0.03	0.03	0.01	0.02	0.01	0.01	0.01	0.03	0.11	0.26	9.52
num_angrys	0.03	0.03	0.04	0.02	0.02	0.01	0.01	0.00	0.01	0.16	11.63

Saaty's CIr values for matrices are given by the following table

Size of Matrix	Random Consistency (CIr)	
1	0	1
2	0	2
3	0.58	3
4	0.90	4
5	1.12	5
6	1.24	6
7	1.32	7
8	1.41	8
9	1.45	9
10	1.49	10

count	9.00
lambda max	14.144
CI	0.643
CR	**0.44**
constant	1.45

Fig. 20. Facebook Status Message Credibility Weights (Model 2).

F. Data Modeling: Proposed Models

In this section, we will illustrate the classification Models performed to classify Users' Profiles into fake or genuine users and Credible or non-credible profiles, as seen in Fig. 21 and 22.

Model 1: Using Supervised Learning "Classification Model" (fake or genuine users).

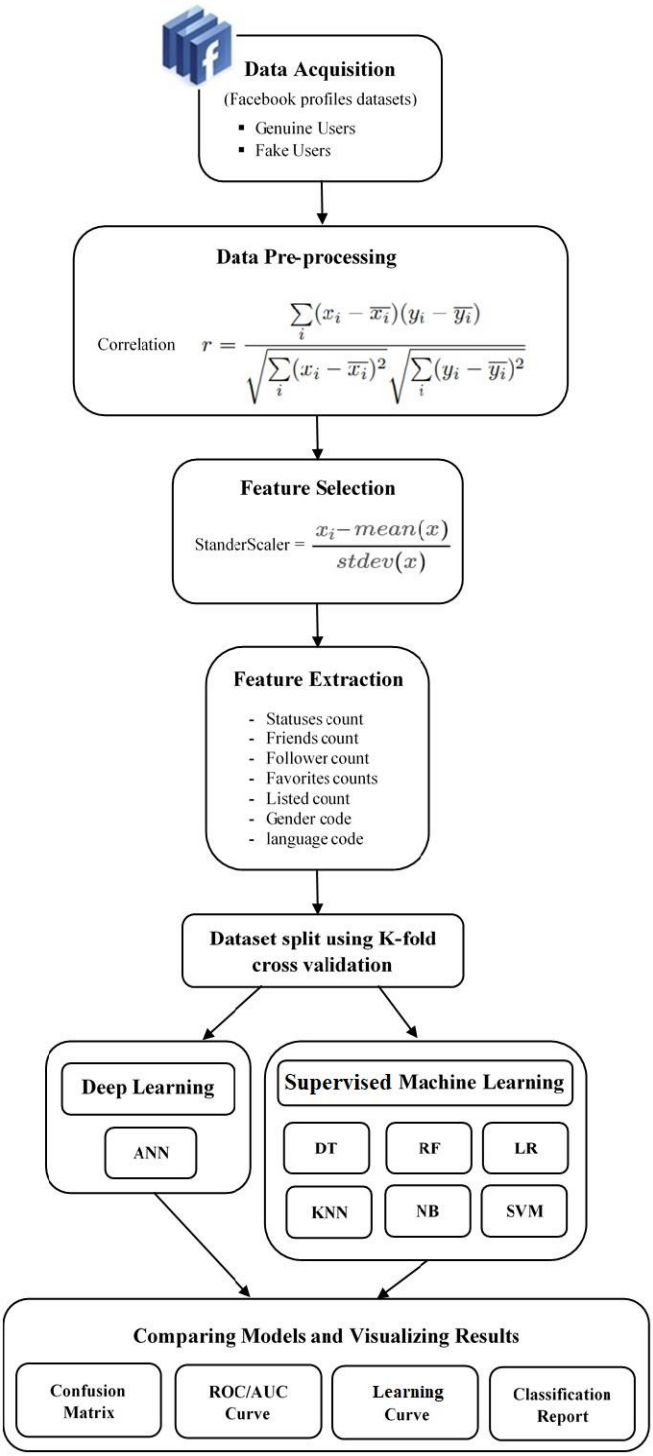

Fig. 21. Fake or Genuine Proposed Model.

Model 2: Using Unsupervised Learning "Clustering Model" (Credible or non-credible profiles).

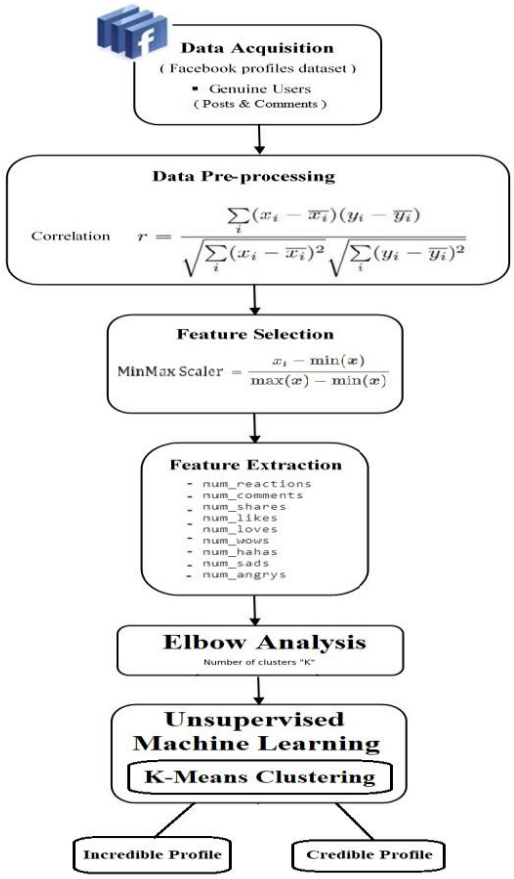

Fig. 22. Credible or Non-Credible Proposed Model.

G. ANN Model: Layers Summary

In our first model, we have built a deep neural network for binary classification to be able to model non-linear relationships and to use Feed-forward neural networks. The model implemented by using, 2 hidden layers with 32 and 16 nodes, using relu activation function. As seen in Fig. 23, the output layer employs the sigmoid activation since it is a binary classification problem. The model achieves 94% training accuracy, pretty well.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 32)                256
_________________________________________________________________
dense_2 (Dense)              (None, 16)                528
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17
=================================================================
Total params: 801
Trainable params: 801
Non-trainable params: 0
_________________________________________________________________
```

Fig. 23. ANN Model: Layers Summary.

V. Experiment Results and Discussion

A. Experiment 1: Discussions

Algorithm 6: Pseudo-code for a Users' profile Binary Classification Model

Input: *Datasets, Classifiers*

Output: *All Models Performance Analysis*

Procedure:

Datasets →{Fake / Genuine, Non-Credible/Credible};
Classifiers →{RF, KNN, SVC, DT, LR, NB, NN};
AllAccuracyScores→{}:
AllRecallScores→{};
AllPrecisionScorcs→{};
Allf1-Scores→{};
AllAUCScores→{};

for DS ∈ Datasets **do**
  for *Xtrain, Xtest ∈ (80%/20% split (DS))* **do**
    Xtrain, Xtest →Perform StanderScaler
    for *clf ∈ Classifiers* **do**
      clf →TrainClassifier(clf, XtrainLabels);
      predictions*→(cls,Xtest);*
      Accuracy→ ComputeAccuracy(predictions,XtestLabels);
      Recall→ ComputeRecall(predictions,XtestLabels);
      Precision→ComputePrecision(predictions,XtestLabels);
      F1-Score→Compute*F1*-Score(predictions,XtestLabels);
      AUC→ComputeAUC(predictions,XtestLabels);
    end for
    return Learning Curves
    return Confusion Matrices (with/without normalization)
    return ROC/AUC Curves
    return Classification Reports
    (AllAccuracyScores, AllRecallScores, AllPrecisionScorcs, Allf1-Scores, AllAUCScores)
    return Fake / Genuine users' profile
    return Non-Credible / Credible users' profile
  end for
end for

1) Discussion on learning curves: For model performance on training and testing, we plot Learning curves that graphs data against varying numbers of training instances. It allows training and testing performance to be viewed separately, to estimate how well models generalize to new data and allow diagnosis of bias and variance problems. High bias is when training/testing errors are high and converge, resulting in poor generalization. High variance is when there is a large gap between the errors, which could indicate there is not enough data or the model is too complex with too many features.

Fig. 24 and 25 represent the learning curve plots for each model in the first experiment. As illustrated in Fig. 24, a neural network learning curve showed a case of a good fit. The training loss plot decreased to a point of stability. Also, the validation loss plot decreased to a point of stability and as noticed from the curve, the gap between both is small.

2) Discussion on confusion matrixes: In the first experiment, the total number of observations that have been labeled was 564 observations in a size of 2x2 matrix according to the binary classification problem.

The following Fig. 26, 27, and 28 shows the confusion matrix for each classification classifier applied on the dataset.

As shown in Fig. 27, the neural network confusion matrix for each of these four values has a specific name. The bottom right is called 'true positives' and indicates that 269 cases, predicted correctly by the classifier, showing the user with a genuine account. The upper left is called 'true negatives' and indicates that 263 cases the classifier correctly predicted the users with a fake account. The upper right is called 'false positives' and indicates that 5 cases only the classifier incorrectly predicted, and the user has a fake account, however, in fact, they do not. The bottom left is called 'false negatives' and indicates that in 27 cases the classifier incorrectly predicted that the user account is genuine when in fact they do have a fake account. We also use the confusion matrix to calculate the accuracy by adding the 'true positives' and the 'true negatives' then dividing them by the total number of observations.

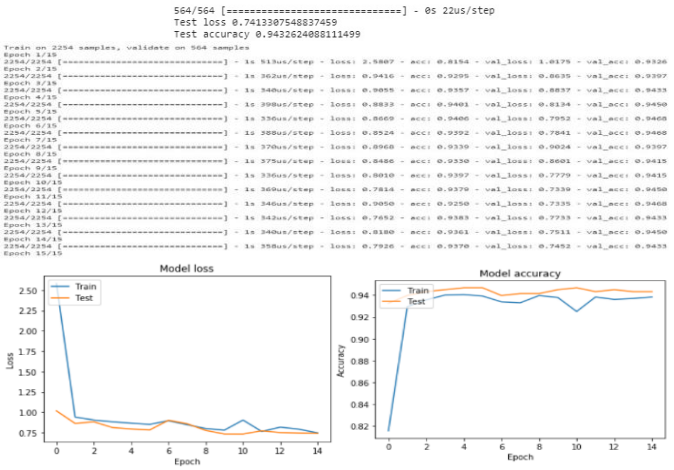

Fig. 24. Neural Network Learning Curve.

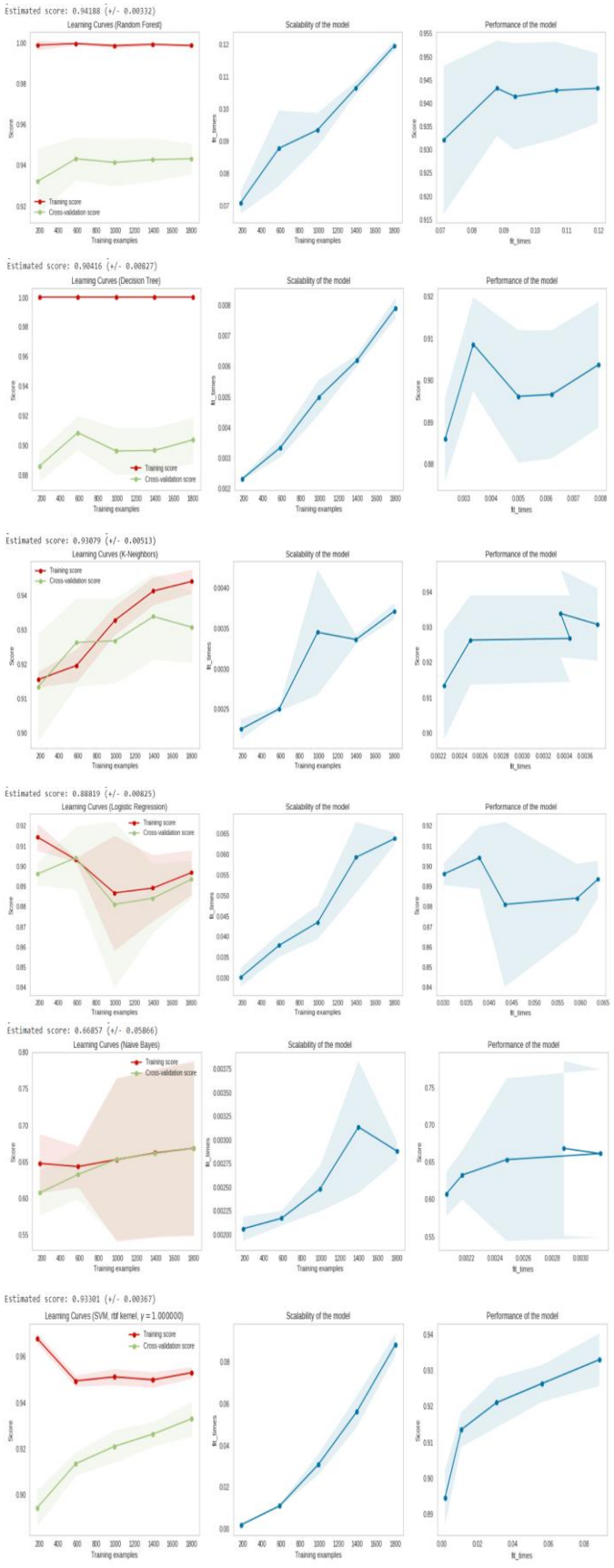

Fig. 25. Summarization for each Model Learning Curves.

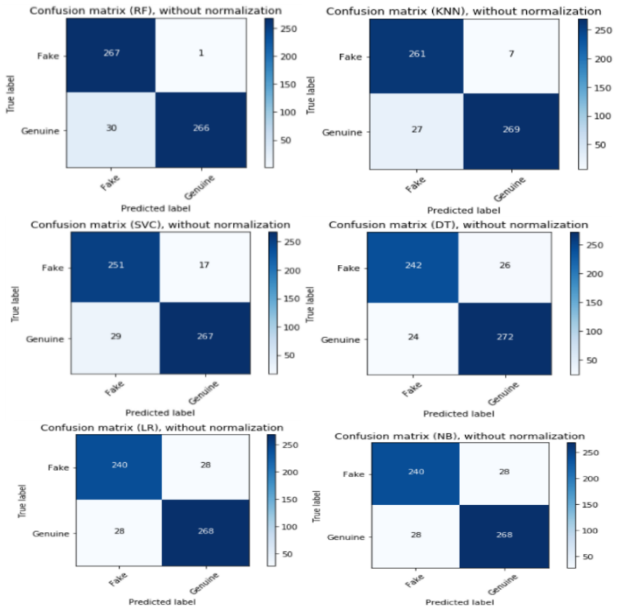

Fig. 26. Summarization for each Model Confusion Matrix (Model 1).

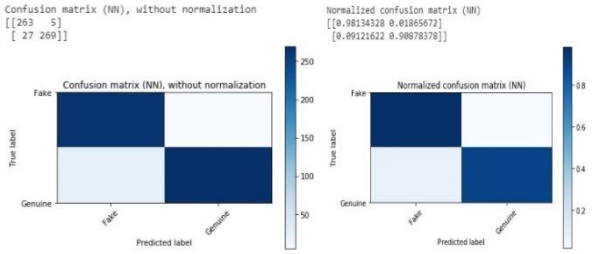

Fig. 27. Neural Network Confusion Matrix (Model 1).

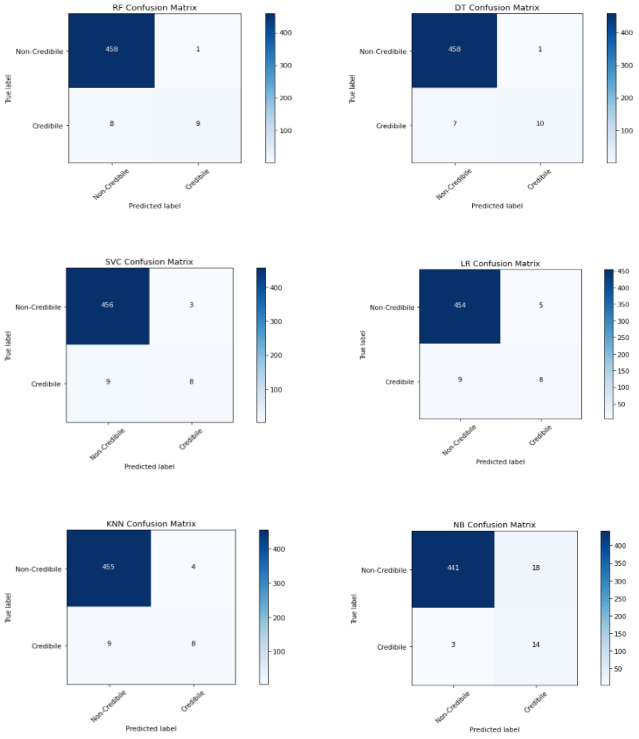

Fig. 28. Summarization for Credibility Confusion Matrix (Model 2).

3) Discussion on ROC/AUC Curves: AUC was calculated for each classifier and used to plot the ROC curve plots to compare the discriminatory powers of the models based on predicted outcome vs. true outcome, as illustrated in the below cumulative, Fig. 29. The ROC curve visualizes the ability to pick a threshold that balances both "sensitivity" and "specificity", to produce the model. Unfortunately, the thresholds cannot be viewed, that used to generate the ROC curve (on the curve itself.)

4) Discussion on models performance: As shown in Fig. 30, we have summarized all the accuracies, precisions, recalls, and f1-scores that had been achieved for each classification model in our binary classification study shown in (Experiment 1).

5) Discussion on credibility score: The best classifier with the best accuracy score in the classification report was the Random Forest classifier, which achieved 95% and the second-best accuracy score computed for the Neural Network model that achieved 94% in classifying users into the fake or genuine.

B. Experiment 2

In this experiment, the dataset had pre-processed using stemming, and stop-lists to vector the sets. We had performed the LDA (Topic Modelling algorithm) to generate the 10 topic vectors and assigned the most relevant topic label. With the generated 10 topic vectors, we had performed PCA to visualize the distribution, created a radar chart to visualize the distribution of sentiment emotion on each topic, and two correlational matrices to visualize the relationship between topics.

We also used the k-Mean clustering algorithm for user analysis and segmentation. Then, we analyzed and grouped users' profiles based on their number of behaviors like 'share' or 'comment' on posts, in addition to the number of sentiment reactions on those posts including 'like', 'love', 'wow', 'haha', 'sad', and 'angry'. This is useful to identify active and inactive users and classify profiles to credible and non-credible profiles, as seen in Fig. 31 and 32.

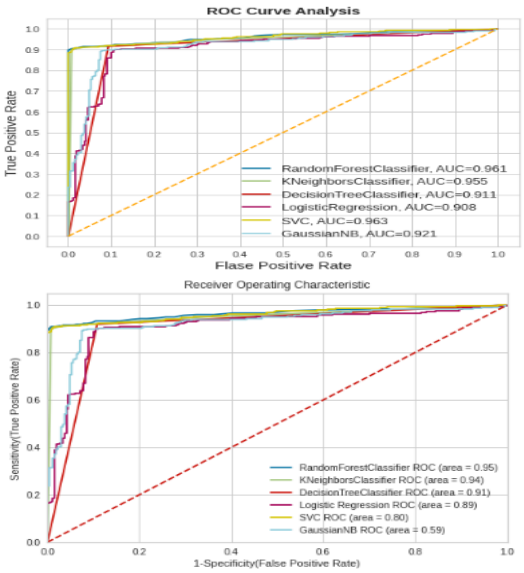

Fig. 29. Cumulative ROC / AUC Curve (Model 1& 2).

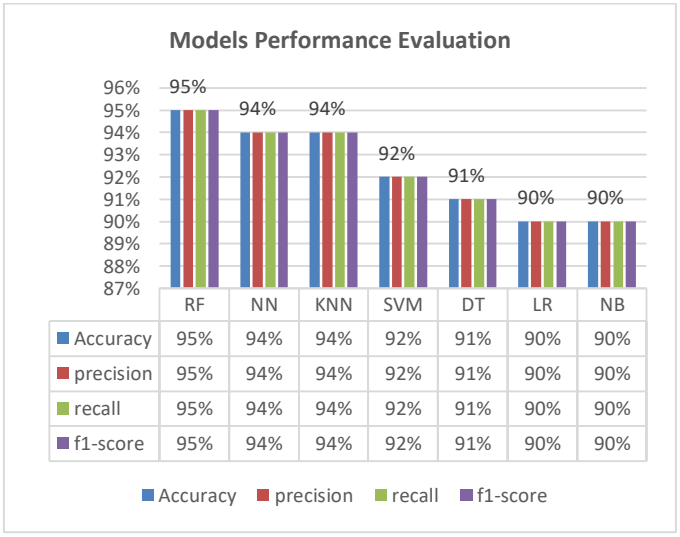

Fig. 30. Models Performance Evaluation (Model 1).

	statuses_count	followers_count	friends_count	favourites_count	listed_count	gender_code	lang_code	Credibility_Score	Label
0	24423	1057	1433	1834	16	0	5	99.83662	Credible
1	24057	1076	840	69	36	0	5	95.76749	Credible
2	22679	560	661	381	7	2	5	89.23221	Credible
3	22540	2065	1125	0	64	0	5	92.91011	Credible
4	22534	715	792	141	2	0	5	89.01435	Credible
...	...	...	...	...	...	...	...	...	...
1442	7	11	49	0	0	-2	5	0.12708	Non-Credible
1443	7	6	54	1	0	2	5	0.12353	Non-Credible
1444	4	0	4	3	0	0	5	0.02516	Non-Credible
1445	4	33	523	0	0	2	5	0.86571	Non-Credible
1446	3	2	4	0	0	0	5	0.02335	Non-Credible

1447 rows × 9 columns

Fig. 31. Exploratory Samples of users' Profile Credibility Score.

	num_reactions	num_comments	num_shares	num_likes	num_loves	num_wows	num_hahas	num_sads	num_angrys	Credibility_Score	Label
0	22364	1347	3829	18512	2717	377	652	45	61	99.96	Credible
1	21460	7654	6934	11469	226	268	1121	2296	6080	99.96	Credible
2	21747	788	6056	19244	1778	541	119	23	42	99.90	Credible
3	20778	1042	11415	15918	985	3676	183	11	5	99.74	Credible
4	21986	4989	4958	13299	1525	317	6542	113	190	99.69	Credible
...	...	...	...	...	...	...	...	...	...	...	...
7962	129	34	0	109	15	4	1	0	0	0.60	Non-Credible
7963	122	44	0	101	6	2	0	13	0	0.58	Non-Credible
7964	66	111	0	65	0	0	1	0	0	0.49	Non-Credible
7965	100	20	0	87	8	1	4	0	0	0.46	Non-Credible
7966	83	40	0	83	0	0	0	0	0	0.43	Non-Credible

7967 rows × 11 columns

Fig. 32. Exploratory Samples of user Reaction Credibility Score.

VI. Conclusion

In this paper, we have implemented two experiments on Facebook user profiles with content generated as posts or comments on pages as CNN page. The first experiment is a binary classification model that automatically detects the fake and genuine profiles. Then, real users classified to credible or not each according to credibility score computed. The researchers computed the credibility score based on the credibility theory using the AHP approach to compute the weights of the correlated features. In this experiment, the Machine Learning and Deep Learning pipeline had been followed. Utilized six supervised machine learning classifiers such as SVM, RF, DT, KNN, LR, NB, and a Deep Learning NN model. The second experiment is a clustering model that classifies the users into two groups of clusters to identify the credible and non-credible users according to their behaviors on posts and comments. In the second model, we had extracted 10 sets of topics by using LDA. After that, we visualized them with sentiments emotions counted from the status message using a correlation matrix to show the dependence and relationship between these various sets. We used the radar charts to plot the 10 sets of topics with sentiment emotions features. Then, we found that the most reactions related to sadness or angry as a negative behavior response related to the time the dataset collected concerning political directions and presidential elections. We had verified the results of the observations for each emotional reaction, then visualized and computed the Principal Component Analysis. In this experiment, we also followed the Machine Learning pipeline using the k-means cluster as an unsupervised learning algorithm to assign each status to the most relevant topic creating the topics sets. And we used the supervised learning algorithms to classify the labels for the topic's sets. In addition to experiment 1, we have plotted the Learning Curves for each model performed to show the model stability. Applied different methods to evaluate the performance of the classifiers, such as the Confusion Matrix table and the ROC/AUC curve. Those two methods described the classifier performance. Implementing both whenever possible, will be beneficial for evaluating any model. The primary characteristic of the Confusion Matrix is the numerous evaluation that can be calculated with it, such as Accuracy Score, Precision Score, Recall Score, and the F1-Score. Also, we can concentrate on the metrics that resemble our research scope. On the other hand, the major characteristic of ROC/AUC curves is, they do not demand us to pick a classification threshold, unlike the Confusion Matrix. We also notice that the main difference between machine learning and deep learning is that deep learning merge's the feature extraction with classification in one process and we don't need to apply the full analysis phase. At the end of this study, experiment '1' results achieved 95% by using the RF classifier and achieved 94% by using the NN model to classify fake and genuine users. Experiment '2' classified the user profiles into credible and non-credible users. This work considered to be the first step that should be performed to measure the profile credibility on Social Media "Facebook" especially status messages with sentiment emotions responses.

References

[1] Gupta, A., & Kaushal, R. (2017). Towards detecting fake user accounts in Facebook. In 2017 ISEA Asia Security and Privacy (ISEASP) (pp. 1-6). IEEE.

[2] Saikaew, K. R., & Noyunsan, C. (2015). Features for measuring credibility on Facebook information. International Scholarly and Scientific Research & Innovation, 9(1), 174-177.

[3] Khedr, Ayman E., N Yaseen, (2017), stock market behavior using data mining technique and news sentiment analysis, International Journal of Intelligent Systems and Applications 9 (7), 22-30.

[4] Khedr, A. E., Idrees, A. M., E Shaaban, E., (2020). Automated Ham-Spam Lexicon Generation Based on Semantic Relations Extraction, International Journal of e-Collaboration (IJeC), 16 (2), 45-64.

[5] Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".

[6] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[7] Harrington, P. (2012). Machine learning in action. Manning Publications Co.

[8] Sultan, Torky I., Khedr, Ayman E., Nasr, Mona M., and Ismail, Walaa S., (2012). Semantic Interoperability-Traditional and Ontology-Based Approaches. In the 1st International Conference of Computing and Informatics (ICCI'2012).

[9] Sharaf Eldin, Ahmed, Khedr, Ayman E., Al-Sharif, Fahad Kamal, (2015), Cross-Language Semantic Web Service Discovery to Improve

the Selection Mechanism by using Data Mining Techniques, International Journal of Computer Applications, 110 (2), 0975 – 8887.

[10] Lê, N. C., Dao, M. T., Nguyen, H. L., Nguyen, T. N., & Vu, H. (2019). An Application of Random Walk on Fake Account Detection Problem: A Hybrid Approach. arXiv preprint arXiv:1911.07609.

[11] Smruthi, M., & Harini, N. (2019). A Hybrid Scheme for Detecting Fake Accounts in Facebook. International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-7, Issue-5S3.

[12] Wani, S. Y., Kirmani, M. M., & Ansarulla, S. I. (2016). Prediction of Fake Profiles on Facebook using Supervised Machine Learning Techniques-A Theoretical Model. International Journal of Computer Science and Information Technologies (IJCSIT), 7(4), 1735-1738.

[13] Yao H., Jiang C., Qian Y. (2019) Introduction. In: Developing Networks using Artificial Intelligence. Wireless Networks. Springer, Cham.

[14] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160, 3-24.

[15] Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems, 83-124.

[16] Tan, P. N., Steinbach, M., & Kumar, V. (2006). Classification: basic concepts, decision trees, and model evaluation. Introduction to data mining, 1, 145-205.

[17] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[18] Fosseng,, S. (2013). Learning Distance Functions in k-Nearest Neighbors (Master's thesis, Institutt for datateknikk og informasjonsvitenskap).

[19] Steinbach, M., & Tan, P. N. (2009). kNN: k-nearest neighbors. The top ten algorithms in data mining, 151-162.

[20] Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[21] Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

[22] Kampakis, S. (2018). What deep learning is and isn't.

[23] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28 (1), 100-108.

[24] Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, 30 (7), 1145-1159.

[25] McGill, R., Tukey, J. W., & Larsen, W. A. (1978). Variations of box plots. The American Statistician, 32 (1), 12-16.

[26] Pearson, K. (1901). Principal Components Analysis. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 6 (2), 559.