Identification of Fake vs. Real Identities on Social Media using Random Forest and Deep Convolutional Neural Network

Bharat Sampatrao Borkar, Rajesh Purohit

Abstract: Abstract—Identity detection is very essential in social media platforms, various platform has facing fake accounts influence since couple of years in current eras. Many researchers has introduces approach for identify the fake profiles, but still system cant able to solve such issues. As these fake identities are being used by offenders for various malicious purposes, it has become necessity of time to identify them. The fake identities are categorized into two main types' i.e. fake identities by bots and fake identities by humans. This system removes fake identities by bots during preprocessing and focuses mainly on identification of fake identities by humans as very little research has been made till now on the fake identities by humans. For classification we test for two different algorithms i.e. Random Forest (RF) and Recurrent Neural Network (RNN). The classification is based on various features such as user name, location, friends count, followers count and so on. Here, dataset used is that of Twitter.

Keywords: social media; identity deception; cyber crimes; machine learning; random forest; deep learning; deep convolutional neural network; activation functions.

I. INTRODUCTION

Social media platforms such as Twitter are one of the most crucial means of communication and information dissemination over internet. Much can be learned about people's behavior by analyzing their profiles on the social media. This helps offenders to create fake identities in order to commit various cyber crimes such as skewing perceptions, manipulation of credit worthiness of accounts, terrorist propaganda, cyber bullying, fraud, identity impersonation, dissemination of pornography, misdirecting people to some malicious website, spreading malwares and so on. These fake identities may be created by bots or humans. The pretend identities by bots typically target giant cluster of individuals at a time, whereas, pretend identities by humans typically target specific individual or restricted variety of individuals. this technique represents Associate in Nursing approach to find pretend identities created by humans on Twitter. In order to classify fake vs. real identities we test for two different machine learning algorithms i.e. Random Forest (RF) and Deep Convolutional Neural Network (DCNN). Furthermore, DCNN is implemented using linear, sigmoid and tan h activation functions. Here, both the algorithms are trained using different cross validation techniques such as 5 fold, 10 fold and 15 fold cross validation. Finally, the system is evaluated based on various performance metrics such as accuracy, precision, recall and F-Measure score in order to predict which activation function as well as cross validation technique gives better classification.

Revised Manuscript Received on October 30, 2019.

Bharat Sampatrao Borkar, Research Scholar, Department of Computer Science & Engineering, School of Engineering & Technology, Suresh Gyan Vihar University, Jagatpura, Jaipur. borkarbharat.sgvu@gmail.com

Dr. Rajesh Purohit, Principal, School of Engineering & Technology, Suresh Gyan Vihar University, Jagatpura, Jaipur. gvset@mygyanvihar.com

II. LITERATURE SURVEY

In machine learning, classification is based on learning from training database. This learning can be categorized into three types as: supervised, semi-supervised and unsupervised. In supervised method of learning class labeled data is present in the beginning. Whereas, in unsupervised learning class labeled data is not available in the beginning. Semi-supervised method of learning is a combination of both supervised and unsupervised learning where some of the class labels are known.

The problem of identification of fake identities can be solved by different classification techniques such as Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Multi Layer Perceptron (MLP), Naïve Bayes (NB), K Nearest Neighbour (KNN), Artificial Neural Network (ANN), Adaboost, Gradient Boosting and so on. Here are some examples,

Estee et. al. [1] trained the classifier using previously used features for bot detection in order to detect fake identities created by humans on Twitter. Here, the classifier is trained exploitation supervised learning technique. they need tested for 3 totally different classifiers i.e. SVM with linear kernel, Adaboost and RF. For SVM, the svm Linear library in R package is employed. Here, the classification boundary is predicated on feature vectors. for enhancing model, Adaboost operate in R package is employed. it's used in conjunction with call trees wherever totally different weight is assigned for every feature so as to predict outcome. These weights square measure changed iteratively so as to gauged effectiveness of classification for every iteration and therefore the method is recurrent till best results achieved. For RF model, RF library within the R package is employed. This model creates variety of trees and mode of sophistication outcome is employed to predict identity deception. Among these three classifiers RF gave the best result.

Sen et. al. [2] used supervised learning method for coaching classifier supported options obtained from FakeLike_data and RandLike_data. they need experimented with numerous classification algorithms like XGBoost, AdaBoost with RF as a base instigator, SVM with RBF kernel, RF, LR and easy feed forward neural network i.e.

MLP to find faux likes on instagram. For MLP two hidden layers with two hundred neurons every area unit used. each layers have sigmoid activation perform and the dropout of output layer is kept 0.2 in order to prevent over fitting. Here, MLP gave better result as compared to other classification techniques.

Sedhai et. al. [3] used semi-supervised learning method in order to coach three totally different classifiers i.e. NB, LR and RF. The classification techniques utilized by these three classifiers are generative, discriminative and call tree primarily based classification model severally. Here, Twitter dataset is employed. Twitter Id is finded as spam on condition that a minimum of two of those three classifiers detect it as a spam. They have called this framework as S^3D (Semi Supervised Spam Detection) and it has reached best classification result compared to any individual classifier.

Xiao et. al. [4] used supervised learning to extract best options from LinkedIn knowledge. they need tested for 3 completely different classifiers i.e. LR with L1 regularization, SVM with radial basis kernel perform and RF a nonlinear tree primarily based ensemble learning methodology. Except regularization LR tries to search out parameters exploitation most chance criterion. In paper L1 penalization is employed to regularize LR model. This methodology maximizes likelihood distribution of sophistication label y given a feature vector x and reduces variety of unsuitable options by exploitation penalty term to certain coefficients of L1 norm. SVM looks for optimal hyperplane as a decision function in high dimensional space. Whereas, RF combines many weak classifiers (decision trees) to form a strong classifier. Here, RF gave best result for identification of fake profiles.

Ikram et. al. [5] used supervised two class SVM classifier enforced discrimination scikit learn (an open supply machine learning library for python) to mechanically distinguish between like farm users from traditional (baseline) users. SVM is compared with alternative accepted supervised classifiers like call tree, AdaBoost, KNN and RF. Here, two class SVM gave best result for identification of like farm users on Facebook.

Dickerson et. al. [6] performed training on Indian Election Dataset (IEDS) extracted from Twitter. they need tested for six high level classifiers like extraordinarily irregular Trees, RF, Gradient Boosting, AdaBoost, Gaussian Naïve Thomas Bayes and SVM. The classifiers were designed and trained on high of scikit-learn, a machine learning toolkit supported by INRIA and Google. Here, AdaBoost performed best on the reduced feature set wherever reduced feature set consists of solely those options that don't involves sentiment analysis. Whereas, Gradient Boosting performed best on full feature set.

Fuller et. al. [7] used dataset provided from law enforcement personal at participating military bases which is also known as "person of interest statements" or Form 1168. Person of interest statements ar official reports written by an issue or witness in a politician investigation. three common classification ways that they need tested ar, ANN, LR and call Tree. Among of these ways ANN reached the most effective performance. ANN could be a assortment of nodes organized in layers. it's three main layers: input layer, hidden layer and output layer. The nodes in hidden layer mix inputs from previous layers into one output price. This output is then passed on to next layer. the burden is related to every unit within the network, it's determined by coaching a network on portion of data. Then network performance is evaluated on holdout sample.

Peddinti et. al. [8] designed a classifier that converts four class classification problem into binary classification problems such that one classifier classifies each account into two classes i.e. anonymous and non anonymous, while, alternative classifies every account as recognizable or non recognizable. Then results of each the classifiers area unit combined so as to classify every account as 'anonymous', 'identifiable' or 'unknown' for Twitter information. each the binary classifiers use RF with one hundred trees as base classifier. the selection of classifier and variety of trees is predicated on cross validation performance and out of bag error. These classifiers area unit value sensitive meta classifiers, where higher cost is imposed for misclassifying instances as anonymous or identifiable.

Oentaryo et. al. [9] used supervised and unsupervised learning methods and tested for four prominent classifiers: NB, RF, SVM and LR. The dataset used is that of Twitter generated by users in Singapore in amount of one Gregorian calendar month to thirty April 2014 and it's extracted via Twitter REST and streaming API. Here, LR gave best result for classification of accounts as Broadcast bots, Consumption bots, Spam bot and Human.

Vishwanath et. al. [10] used unsupervised method of learning for Facebook dataset. The classification is performed using KNN algorithm. In KNN data is classed supported majority vote of its neighbors, with take a look at information being assigned to a category commonest among its k nearest neighbors wherever k could be a positive number generally tiny in worth. Here, classification is completed into four categories i.e. Black market, Compromised, Colluding and Unclassified.

From this literature survey we found that Random Forest and Neural Networks are giving best results for identification of fake profiles on social media. Thus, we test for these two classification techniques in our system.

III. SYSTEM ARCHITECTURE

The flow of our system is as follows:

Data Acquisition:

First of all, data is extracted from Twitter using Twitter API based on keywords such as "school" and "homework" as these are the keywords that are mostly used by minors and minors are more susceptible to cyber crimes. Here we have extracted about 3000 accounts from Twitter.

Preprocessing:

The various preprocessing steps that we have applied are,

Lexical analysis:

Lexical analysis separates the input alphabet into,

a) Word characters: For e.g., letters a-z and

b) Word separators: For e.g., space, newline, tab

Stopword removal:

Stopword removal refers to the removal of words that occur most frequently in the documents. The stopwords includes,

a) Articles (a, an, the,…)

b) Prepositions (in, on, of,…)

c) Conjunctions (and, or, but, if,…)

d) Pronouns (I, you, them, it,…)

c) Possibly some verbs, nouns, adverbs, adjectives (make, thing, similar…)

Stemming:

Stemming replaces all the variants of a word with a single stem word. Variants include plurals, gerund forms (ing forms), third person suffixes, past tense suffixes, etc. Here we used the Porter's algorithm for stemming.

Index term selection:

Index term selection refers to the selection of appropriate features from large amount of data that contribute most to our prediction variable or output.

Data cleaning:

During data cleaning step bots are removed from the dataset based on certain parameters such as presence of name, profile image, number of followers, number of tweets, use of punctuation etc. Also, accounts of known celebrities are removed from the given corpus.

Create fictitious accounts:

Then fictitious accounts are created with the help of various random human data generator APIs and manually by us. The number of fictitious accounts created by us is around 4000. The basis for creation of fictitious accounts is that the people generally lie on their age, gender, image, location and the name most. For example, if location given is that of Arctic ocean or some volcano where human being cannot survive then it can be considered as fake.

Validate data:

For the sake of validity of research, it is decided to ensure that the fabricated deceptive accounts are as far as possible aligned with the accounts extracted via Twitter API in data acquisition step (original corpus). This was done to make our research results as realistic as possible. For that we have implemented following two statistical tests,

- Mann-Whitney U test:

  This test proofs that the means of the two sets are similar per attribute.

- Chi square test:

  This test proofs that the datasets are not correlated and therefore independent.

This means that both the deceptive and original corpus must have similar data and show same distributions.

Inject fictitious accounts:

Fictitious accounts that pass Mann Whitney U test and Chi square test are injected into the system. Thus, now our corpus will consists of fake and real accounts by humans extracted via Twitter API as well as the fake accounts that we have created manually and the total number of accounts becomes about 7000.

Create new features:

Here some new features are created using features that we have extracted in preprocessing step which made identification of fake identities much easier. For example, ratio of tweets containing URL to the total number of tweets is higher for fake identities as the URLs are used by offenders to misdirect people to malicious websites.

Classification:

We have tested for two different algorithms i.e. Random Forest and Deep Convolutional Neural Network (Linear, Sigmoid and Tan h activation function) for classification of Fakes vs. Real identities. Both the algorithms are trained using supervised learning method. Here we have experimented with three different cross validation techniques i.e. 5 fold, 10 fold and15 fold where 70 percent data is given for training and remaining 30 percent data goes for testing.

Random Forest:

a) Algorithm:

1) Randomly select k features from total m features, where k is less than m in order to construct n decision trees.

2) Take the test vector and use rules of each randomly created decision tree to predict the outcome and then store predicted outcome.

3) Calculate the votes for each predicted outcome.

4) Consider the highly voted predicted outcome as the final prediction of random forest algorithm.

c) Activation function:

$$W = \sum_{i=0}^{n}(inp[i]) = (hid[i]) \quad (1)$$

W > T: 1;

W < T: 0

Where,

W is a weight assigned based on equality of input and hidden identities.

Here, input (inp) corresponds to the identities whose class label is to be detected and hidden (hid) identities corresponds to the training data whose class label is known.

T is a threshold kept on calculated weight to detect fake identities.

Deep Convolutional Neural Network:

a) Algorithm:

1) First of all we inject number of Twitter accounts that we have extracted via Twitter API to the system for classification purpose. Now Input layer consists of all samples like I = (input sample 1, input sample 2,….., input sample n).

2) Now first convolution layer is dependent on training database which can generate the output samples based on current classification weight which will be given as a input to next layer.

3) Then second convolution layer is dependent on background knowledge i.e. classification rules. The output samples of this layer are then provided to output layer where different activation functions can be applied on it for classification purpose.

4) Finally output layer gives the final output labeled in the form O = (Fake accounts, Real accounts). During whole process it follows Feed Forward architecture.

c) Activation functions:

For DCNN we have tested for three different action functions i.e. Linear, Sigmoid and Tan h.

1) Linear activation function:

y = a + v

Where,

v = $\sum w_i x_i$

x_i is a set of features.

w_i are weights associated with features.

a is bias.

2) Sigmoid activation function:

$y = 1/(1 + e^{-v})$ (3)

Where,

v = $\sum w_i x_i$

x_i is a set of features.

w_i are weights associated with features.

3) Tan h activation function:

y = tanh(x) = $2/(1 + e^{-2v}) - 1$ (4)

Where,

v = $\sum w_i x_i$

x_i is a set of features.

w_i are weights associated with features

Evaluate results:

Results are evaluated based on various performance metrics such as accuracy, precision, recall and f1 score.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

F1 score = 2 * (Precision * Recall)/(Precision + Recall) (8)

Where,

TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

Recurrent Neural Network

Input: Test Dataset which contains various test instances TestDBLits [], Train dataset which is build by training phase TrainDBLits[] , Threshold Th.

Output: HashMap <class_label, SimilarityWeight> all instances which weight violates the threshold score.

Step 1: For each read each test instances using below equation

$$testFeature(m) = \sum_{m=1}^{n} (.\ featureSet[A[i] \ldots\ldots A[n] \leftarrow \text{TestDBLits})$$

Step 2 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using below equation.

Extracted_FeatureSetx[t……n] = $\sum_{x=1}^{n}(t) \leftarrow testFeature$ (m)

Extracted_FeatureSetx[t] contains the feature vector of respective domain

Step 3: For each read each train instances using below equation

$$trainFeature(m) = \sum_{m=1}^{n} (.\ featureSet[A[i] \ldots\ldots A[n] \leftarrow \text{TrainDBList})$$

Step 4 : extract each feature as a hot vector or input neuron from $testFeature(m)$ using below equation.

Extracted_FeatureSetx[t……n] = $\sum_{x=1}^{n}(t) \leftarrow testFeature$ (m)

Extracted_FeatureSetx[t] contains the feature vector of respective domain.

Step 5 : Now map each test feature set to all respective training feature set

$$weight = calcSim\ (\text{FeatureSetx} \mid\mid \sum_{i=1}^{n} \text{FeatureSety[y]})$$

Step 6 : return instance[label] [weight]

Random Forest

Input : Selected feature of all test instances D[i….n], Training database policies {T[1]……T[n]}

Output: No. of probable classified trees with weight and label.

Step 1: for each (D[i] into D)

Select n attributes randomly from D[i] using below formula

$$Treeset[k] = \sum_{k=0}^{n} attribute[D[i]k \ldots D[n]n]$$

Step 2: for each (T[i] into T)

$$Train[m] = \sum_{m=0}^{n} attribute[T[i]m \ldots T[n]n]$$

Step 3: calculate weight between train and test instance

$$Treeset[k].weight = similarity\ (Treeset[k] \sum_{i=1}^{n} Train[m]$$

Step 4: $if(Treeset[k].weight \geq Th)$ $Treeset[k].label \leftarrow Train[m].class$

Break;

Step 5: return $Treeset[k].label$

IV. RESULTS AND DISCUSSION

Table 1: Results for Fake vs. Real Identities Classification on Social Media using5 fold and 10 fold and 15 fold cross validation training of Random Forest (RF) Algorithm.

RF	FOLD 5	FOLD 10	FOLD 15
ACCURACY	85.40	86.10	89.40
PRECISION	84.40	86.50	89.40
RECALL	85.70	86.65	89.70
F1 SCORE	86.50	86.90	89.50

The above table 1 shows classification accuracy of Random Forest with 5 fold, 10 fold and 15 fold respectively. Basically around 3000 account initial input data has given for classification, execute the train and test module respectively. It provides around 89.40% accuracy for 15 fold while 85.50% accuracy for 5 fold splitting the data.

V. CONCLUSION

The maximum accuracy with which problem of classification of fake vs. real identities on social media can be highest and it is achieved by DCNN with different activation function. The performance of given system varies with dataset used for it. Also we found that the classification accuracy of system increases as the number of folds used in system increases.

REFERENCES

1. Estee Van Der Walt and Jan Eloff, "Using Machine Learning to Detect Fake Identities: Bots vs Humans," IEEE, 2018
2. Indira Sen et. al. "Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram," ACM,2018.
3. B. Viswanath et. al. "Towards Detecting Anomolous User Behaviour in Online Social Networks," USENIX, 2014.
4. Surendra Sedhai and Aixin Sun, "Semi-Supervised Spam Detection in Twitter Stream," IEEE, 2018.
5. Cao Xiao, David Freeman and Theodore Hwa, "Detecting Clusters of Fake Accounts in Online Social Networks," ACM, 2015.
6. Ikram et. al., "Combating Fraud in Online Social Networks: Detecting Stealthy Facebook Like Farms," ARXIV, 2016
7. J. Dickerson,V. Kagan and V. Subhramanian,"Using Sentiment to Detect Bots on Twitter: Are Humans more Opinionated than Bots?," IEEE,2014.
8. C. Fuller, D. Biros and R. Wilson "Decision Support for Determining Veracity via Linguistic based Cues," ELSEVIER, 2009.
9. S. Peddinti, K. Ross and J. Cappos "Mining Anonymity: Identifying Sensitive Accounts on Twitter," ARXIV, 2016.
10. R. Oentaryo et. al. "On Profiling Bots in Social Media," ARXIV, 2016.
11. Manuel Egele, Gianluca Stringhini, Christopher Kruegel, Giovanni Vigna "Towards Detecting Compromised Accounts on Social Networks," IEEE, 2017.

AUTHORS PROFILE

Mr. Bharat Sampatrao Borkar did B.E in Computer engineering from Pune University in 2004 and M.E in Computer Science & Engineering from M.G.M's College of Engineering, Nanded (S.R.T.M. University) in 2009. He is Currently pursuing Ph.D in Computer Science & Engineering from Suresh Gyan Vihar University Jaipur (Rajasthan). His research field area is Machine Learning.

Dr. Rajesh Purohit did Electrical Engineering with specialization in Control of Electrical Machines from BITS Pilani in 1993 and M.S from BITS Pilani in 2009. He had done Ph.D. Birla Institute of Technology & Science, Pilani Campus in 2014. He is presently working as Principal of School of Engineering & Technology at Suresh Gyan Vihar University Jaipur (Rajasthan).