IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

KOLLECTOR: Performance Evaluation of Several Classification Models for Fraudulent Activities on Mobile Devices

BURRI ANIL KUMAR [#1], MADUGULA MURALI KRISHNA[#2] ,

[#1] M.Tech Student, Department of Computer Science and Engineering,

[#2]Assistant Professor, Department of Computer Science and Engineering,

Sri Sivani Colleges of Engineering, Chilakapalem Junction, Srikakulam-532402.

ABSTRACT

Preventing privacy and personal information leaks has become a difficult job as smartphone usage has grown quickly. Impersonation is one of the main effects of such leaks. Due to the inability of current preventative measures (such as passcodes and fingerprints) to continually monitor usage and determine if a user is permitted, this kind of unlawful usage is practically hard to stop. Unauthorized users would have complete access to the devices after they were able to circumvent the first security measures, including utilising passwords they had previously stored to get into valuable websites. We introduce KOLLECTOR, a brand-new framework for detecting impersonation that uses a multi-view bagging deep learning method to gather data on successive tapping on a smartphone's keyboard. For continuous user authentication while typing, we build a sequential-tapping biometrics model. We assessed statistically. The results of our comparison between our model and widely established shallow machine techniques show that our system outperforms other methods and can reach an 8.42% equal error rate, a 94.24% accuracy, and a 94.41% H-mean using only the accelerometer and five keyboard taps. We also test the method with only three keyboard taps and discover that it still produces excellent accuracy while providing more possibilities to make more decisions that might lead to more accurate final judgments.

Keywords: KOLLECTOR, Fingerprints, Smartphone, Deep Learning Models, Classification, Authentication.

1. INTRODUCTION

Smart mobile device proliferation has made it possible for consumers to "computing anytime, anywhere." However, because these smart mobile gadgets hold just as much private and sensitive information as less mobile equipment like desktops and laptops, their mobility has also made them targets for theft. More than three million Americans fell victim to smartphone theft in 2013 [1]. When these devices are stolen, electronic impersonation is frequently used to access personal data by pretending to be the owner of the device. The major issue of impersonation fraud cost U.S. businesses close to $180 million between 2013 and 2014 [2]. Studies have revealed that most owners select relatively basic passwords or even no access codes, despite the fact that setting access codes can protect these devices.Numerous studies have also demonstrated that even when passcodes are used, hackers may still decipher passwords by looking for tap, fingerprint, and/or smudge patterns on displays [6, 7, 8, 9]. Existing security measures are notable for merely attempting to prevent unauthorised users from unlocking devices. There are no more safeguards to keep them from utilising the gadget when they manage to get through these ones. In order to prevent unauthorised use, it is highly desirable to improve the authentication mechanisms in smart phones to include additional defensive measures designed to be non-intrusive but capable of continuously monitoring user activity such as web browsing or entering information into web applications.

We decided to build a continuous identification system in response to these problems, subject to the following design specifications. The system must first function in an environment known as the "open world" where potential attackers are not present. A new cross-validation mechanism is needed for this. Second, the system should be able to recognise the user of the device with high accuracy and a low equal error rate after just a few keyboard inputs. Third, the system must be rapid at doing the categorization while utilising the fewest possible smartphone sensors and keeping them active for the shortest amount of time possible.

In this article, we review various articles that have employed one or more data mining methods to identify cyberbullying. Studies of the results using various ML algorithms show that nearly no publication ever achieves 100% accuracy. As a result, some sets of ML algorithms are unable to accurately identify cyberbullying, and in this research, we seek to increase the accuracy of the corresponding techniques.

In this system, we attempt to employ a number of ML algorithms to identify texts that have been cyberbullied from online social networks. Here, we examine the effectiveness of several machine learning (ML) algorithms in identifying cyberbullying texts from a set of talks gathered from online social networks

2. LITERATURE SURVEY

1) "Personal identification based on iris texture analysis," IEEE transactions on pattern analysis and machine intelligence, vol. 25, no. 12, pp. 1519–1533, 2003. L. Ma, T. Tan, Y. Wang, and D. Zhang.

Automated personal identification based on biometrics has received a lot of attention over the past 10 years due to the increased emphasis on security. Iris recognition is a new biometric recognition technique that is gaining a lot of attention in academic circles and in real-world applications. Iris imaging, liveness detection, and identification are all often included in a standard iris recognition system. The last problem is the main topic of this work, which also introduces a novel method for iris detection from a picture series. Prior to choosing a clear iris picture for later recognition, we first evaluate the quality of each image in the input sequence. a collection of spatial filters with iris recognition-friendly kernels, is then applied to extract distinguishing texture features by capturing local iris properties. According to experimental findings, the proposed technique performs admirably. A comparison of existing iris identification techniques is specifically done using a library of 2,255 iris picture sequences from 213 people. Conclusions drawn from such a comparison using the bootstrap, a nonparametric statistical approach, offer helpful data for additional study.

2) Recurrent neural network regularization, W. Zaremba, I. Sutskever, and O. Vinyals, arXiv preprint arXiv:1409.2329, 2014.

We offer a straightforward regularization method for recurrent neural networks (RNNs) equipped with LSTM units. The most effective regularization method for neural networks, dropout, does not perform well with RNNs and LSTMs. In this study, we demonstrate the proper use of dropout to LSTMs and demonstrate that it significantly lowers over fitting on a range of tasks. Language modeling, speech recognition, creating picture captions, and machine translation are some of these responsibilities.

3) A Convolutional Neural Network Approach for Objective Video Quality Assessment, IEEE Transactions on Neural Networks, P. Le Callet, C. Viard-Gaudin, and D. Barba, 2015

The objective measuring approach used in this paper's use of neural networks to automatically evaluate the perceived quality of digital movies is described. This complicated problem seeks to replace subjective quality evaluation, which is highly time-consuming and complex. To address this issue, several criteria have been put out in the literature. They are built around a broad framework that incorporates many stages, each of which deals with difficult issues. This paper's goal is to explore a novel use of neural networks in such a framework in the context of the reduced reference (RR) quality metric, rather than to propose a worldwide ideal quality measureWe specifically draw attention to the use of such a tool for pooling and combining information in order to calculate quality ratings. The suggested technique addresses a few issues with objective measures that should be able to forecast the subjective quality score achieved by the single stimulus continuous quality assessment (SSCQE) method.

3. EXISTING SYSTEM & ITS LIMITATIONS

There was no pre-established mechanism or programme in the current system to detect online fraud operations. Therefore, the following are the limits of the current system. They are listed below.

1. All of the current systems make an effort to detect fraud manually.
2. Using a manual technique to identify fraudulent activity is inaccurate.
3. Tracing the fraud operations is ineffective.
4. If the dataset is little, it takes less time to discover fraud activities; nevertheless, if the dataset is vast, it is a very difficult process to do so.

4. PROPOSED SYSTEM & ITS ADVANTAGES

For feature learning, ML techniques were employed in the suggested system. The suggested system uses an ML-Approach to categories all actions and track fraud and regular activity separately. The suggested technique is also quite effective in locating the underlying source of that fraudulent conduct.

ADVANTAGES OF THE PROPOSED SYSTEM

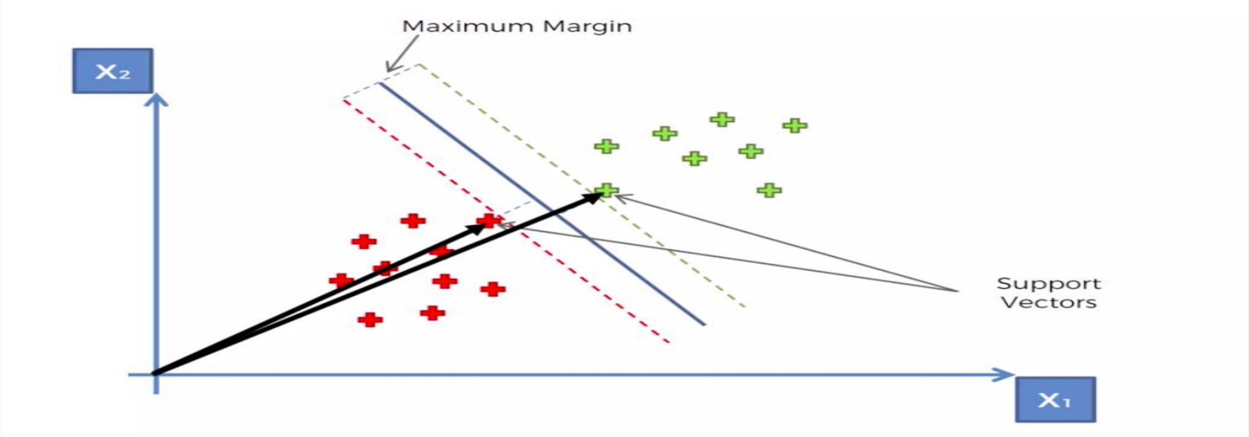
1. In the proposed system, fraud actions on any dataset may be automatically classified.
2. The suggested strategy is quite precise
3. It is incredibly effective and simple to categories.
4. This strategy makes it simple to categorize fraud activity in both small and large datasets.

5. PROPOSED MODELS

Here we used some classification models for obtaining the performance of best classification model to identify the fraudulent users from mobile phones.

1) Support Vector Machine

A supervised machine learning technique called the Support Vector Machine (SVM) may carry out classification, regression, and even outlier identification. The linear SVM classifier functions by connecting two classes using a straight line. All of the data points that fall on one side of the line will be given a single class label, and all of the points that fall on the other side will be given a second class label. Although it seems straightforward, there are an endless number of lines to pick from. How can we determine which line will classify the data the most accurately? The LSVM method is useful in this situation. In addition to separating the two classes, the LSVM algorithm will choose a line that is as far away from the.



2) Decision Tree Classification

How to construct and tune Decision Tree Classifiers using the Python Scikit-learn module, as well as attribute selection metrics. You want a group of clients who are most likely to buy your goods as a marketing manager. By identifying your audience, you may reduce your marketing expenditures. To reduce the rate of loan defaults, you must recognise hazardous loan applications as a loan manager. A classification difficulty occurs when clients are divided into groups of prospective and non-potential customers or safe or dangerous loan applications. The two steps of classification are learning and prediction. The model is created using provided training data in the learning phase. The model is applied to forecast in the prediction stage..

3) Random Forest Algorithm

A supervised machine learning approach based on ensemble learning is known as random forest. In order to create a more effective prediction model, you can combine several kinds of algorithms or use the same technique more than once in ensemble learning. The term "Random Forest" comes from the fact that the random forest method mixes several algorithms of the same type, or different decision trees, into a forest of trees. Both regression and classification tasks may be performed using the random forest approach.

6. IMPLEMENTATION PHASE

The step of implementation is when the theoretical design is translated into a programmatically-based approach. The application will be divided into a number of components at this point and then programmed for deployment. The application's front end uses Google Collaboratory, while for the back end database, we used the dataset ClaMP Integrated-5184.csv. Python is being used in this instance to implement the present application. The following six modules make up the bulk of the application. They are listed below:

1. Initialize Dataset

2. Data preparation

3. Visualizations of Data

4. Use Machine Learning algorithms to find fraudulent activities

5. Perform a performance analysis and calculate accuracy.

Now let us discuss about each and every module in detail as follows:

1) LOAD DATASET MODULE:

In order to use this module, we must first load the input dataset, which is made up of a collection of pre-defined actions gathered from mobile phones. The dataset is downloaded from the KAGGLE website to assess the effectiveness of the suggested application..

2) DATA PRE-PROCESSING MODULE

In this module, we attempt to pre-process the data using the Python library's Natural Language Tool Kit module. In this case, we attempt to split the entire dataset into two segments: testing and training, and then attempt to construct the entire dataset for both test and train.**3.2.3. DATA 3) VISUALIZATION MODULE**

Here, the system-uploaded data is shown to determine how many properties in total are included in the dataset.

4) APPLY ML ALGORITHMS MODULE

Here, we aim to employ a variety of machine learning (ML) methods to spot fraudulent activity, including SVM, Decision Trees, Neural Networks, and Random Forest.

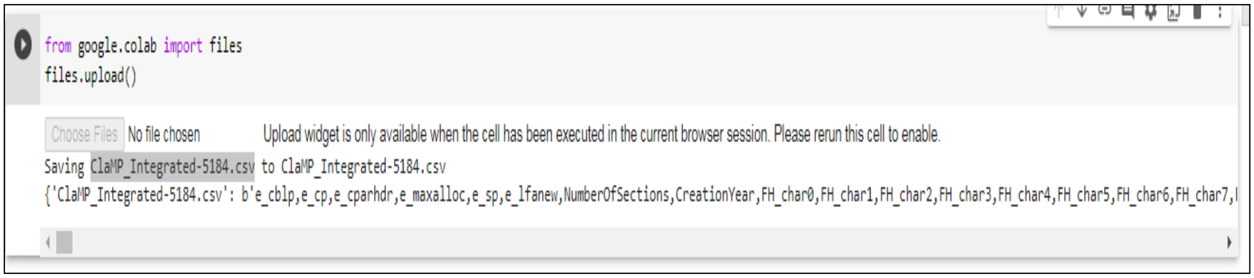
5) CALCULATE ACCURACY AND REPORT GENERATION

We examined the dataset using SVM, Decision Trees, Neural Networks, and the Random Forest algorithm in the current application. Finally, we got to the conclusion that Random Forest provides the highest accuracy for detecting mobile phone fraud.

7. EXPERIMENTAL REPORTS

Here we used google collab as working environment to show the performance of our proposed application and for developing the application we used python as programming language. Now we can see the experimental reports for implementing the proposed application on multiple classification algorithms.

1) Sample Code for Loading Dataset

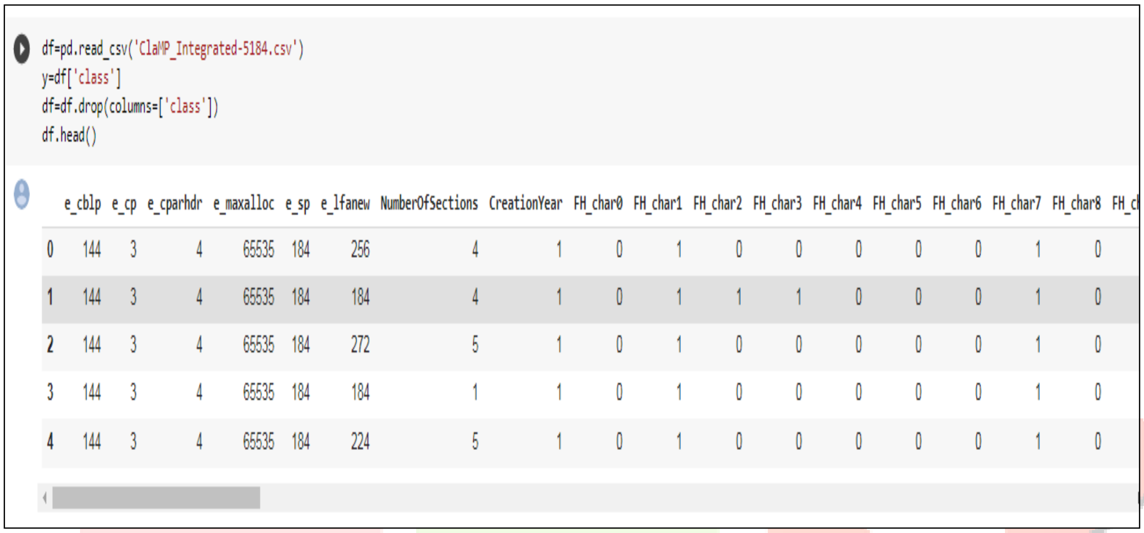


```
from google.colab import files
files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving ClaMP_Integrated-5184.csv to ClaMP_Integrated-5184.csv
{'ClaMP_Integrated-5184.csv': b'e_cblp,e_cp,e_cparhdr,e_maxalloc,e_sp,e_lfanew,NumberOfSections,CreationYear,FH_char0,FH_char1,FH_char2,FH_char3,FH_char4,FH_char5,FH_char6,FH_char7,
```

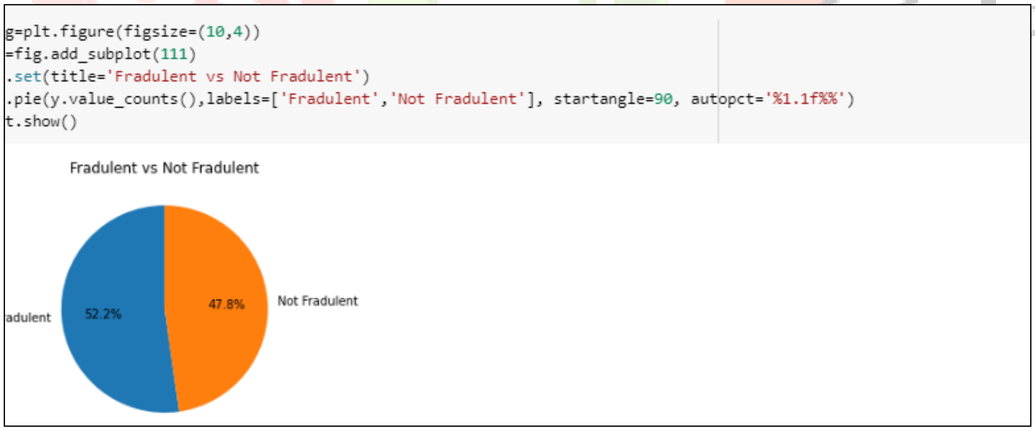
2) Dataset Loaded



```
df=pd.read_csv('ClaMP_Integrated-5184.csv')
y=df['class']
df=df.drop(columns=['class'])
df.head()
```

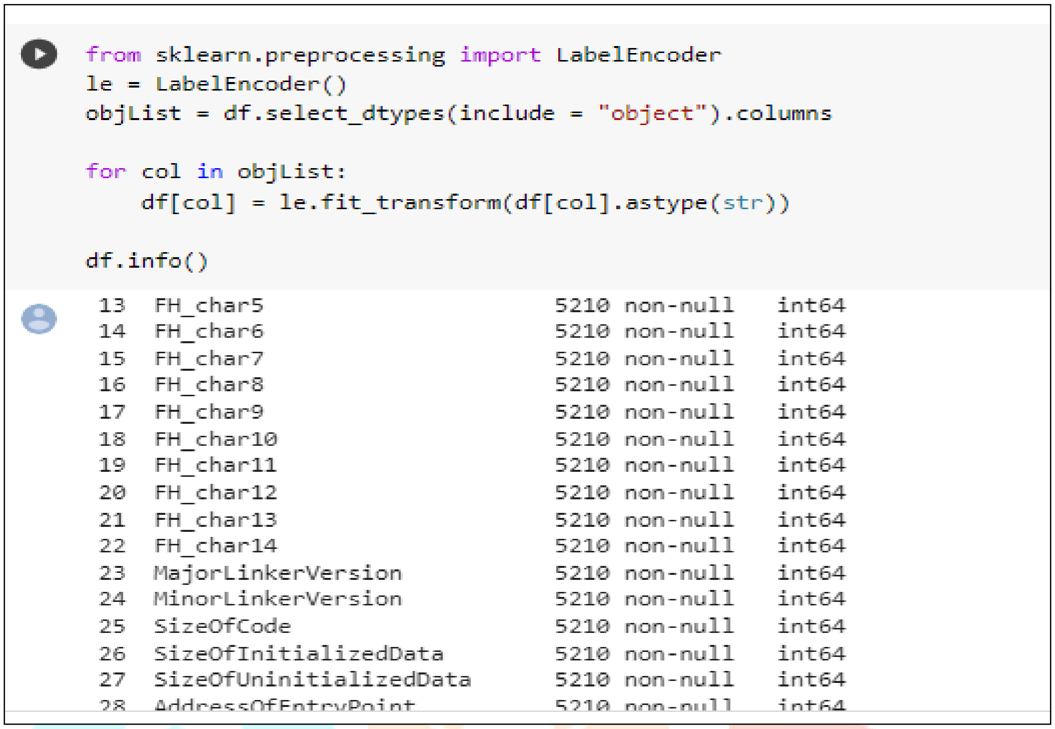
	e_cblp	e_cp	e_cparhdr	e_maxalloc	e_sp	e_lfanew	NumberOfSections	CreationYear	FH_char0	FH_char1	FH_char2	FH_char3	FH_char4	FH_char5	FH_char6	FH_char7	FH_char8
0	144	3	4	65535	184	256	4	1	0	1	0	0	0	0	0	1	0
1	144	3	4	65535	184	184	4	1	0	1	1	1	0	0	0	1	0
2	144	3	4	65535	184	272	5	1	0	1	0	0	0	0	0	1	0
3	144	3	4	65535	184	184	1	1	0	1	0	0	0	0	0	1	0
4	144	3	4	65535	184	224	5	1	0	1	0	0	0	0	0	1	0

3) Data Pre-Processing



```
g=plt.figure(figsize=(10,4))
=fig.add_subplot(111)
.set(title='Fradulent vs Not Fradulent')
.pie(y.value_counts(),labels=['Fradulent','Not Fradulent'], startangle=90, autopct='%1.1f%%')
t.show()
```

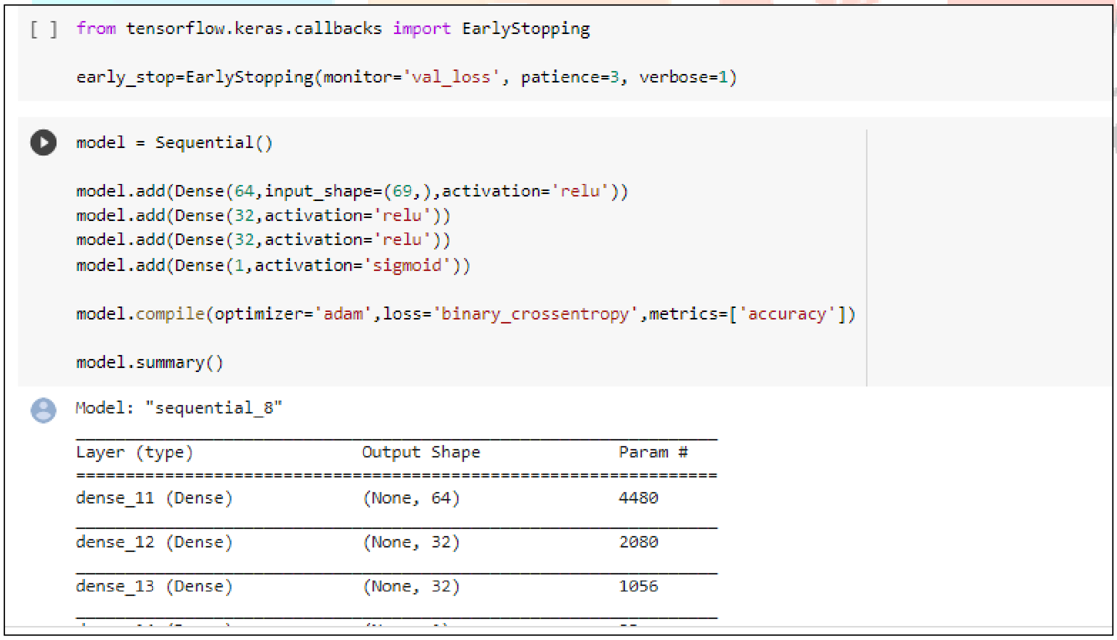
4) Install the Respected Packages



```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
objList = df.select_dtypes(include = "object").columns

for col in objList:
    df[col] = le.fit_transform(df[col].astype(str))

df.info()
```

```
 13  FH_char5                     5210 non-null   int64
 14  FH_char6                     5210 non-null   int64
 15  FH_char7                     5210 non-null   int64
 16  FH_char8                     5210 non-null   int64
 17  FH_char9                     5210 non-null   int64
 18  FH_char10                    5210 non-null   int64
 19  FH_char11                    5210 non-null   int64
 20  FH_char12                    5210 non-null   int64
 21  FH_char13                    5210 non-null   int64
 22  FH_char14                    5210 non-null   int64
 23  MajorLinkerVersion           5210 non-null   int64
 24  MinorLinkerVersion           5210 non-null   int64
 25  SizeOfCode                   5210 non-null   int64
 26  SizeOfInitializedData        5210 non-null   int64
 27  SizeOfUninitializedData      5210 non-null   int64
 28  AddressOfEntryPoint          5210 non-null   int64
```

5) Apply Models

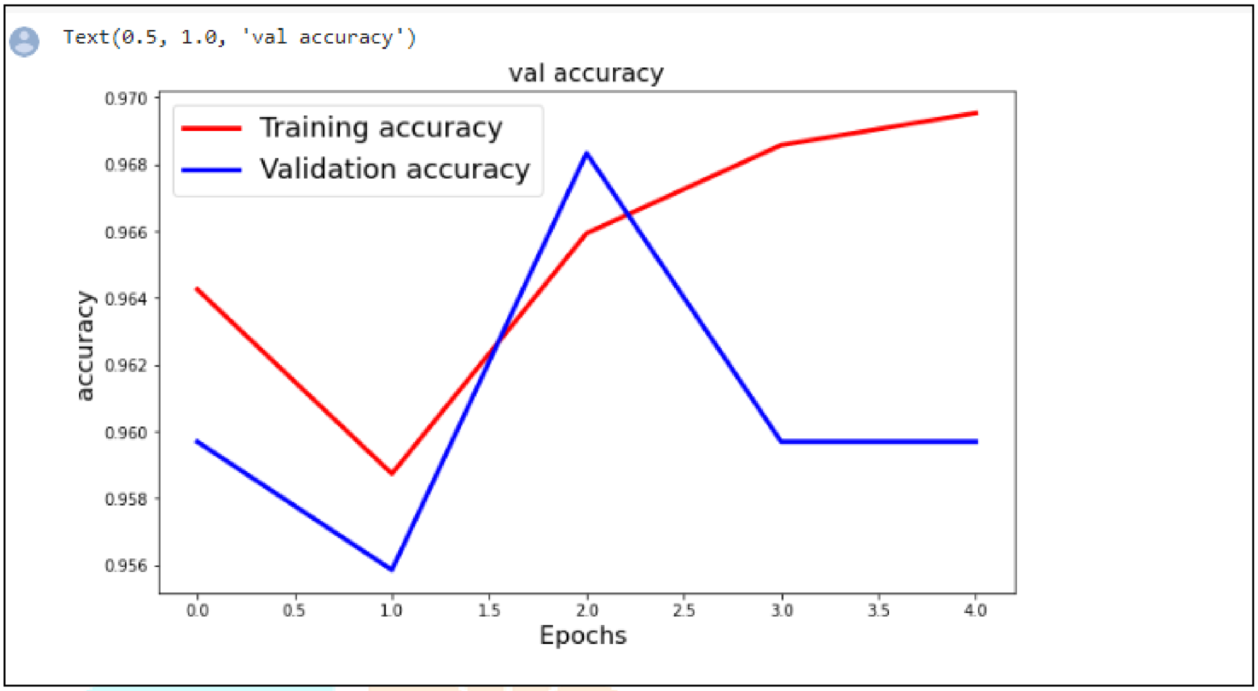


```
[ ] from tensorflow.keras.callbacks import EarlyStopping

    early_stop=EarlyStopping(monitor='val_loss', patience=3, verbose=1)
```

```
model = Sequential()

model.add(Dense(64,input_shape=(69,),activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

model.summary()
```

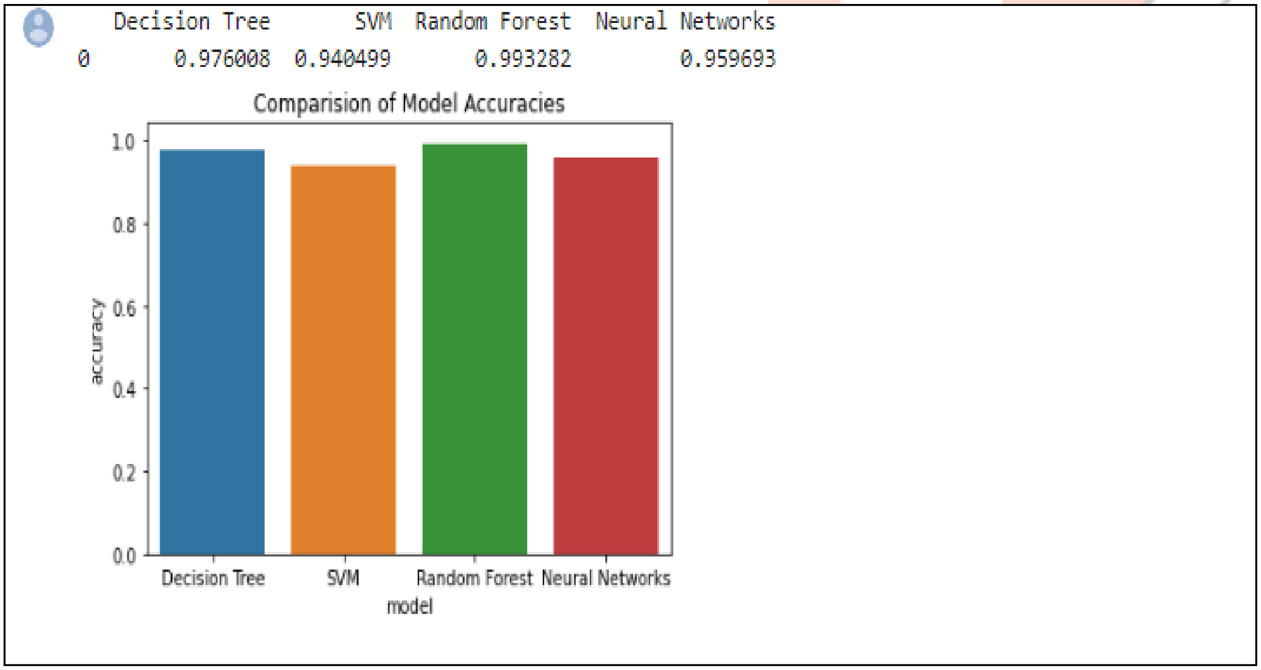
```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
dense_11 (Dense)            (None, 64)                4480

_________________________________________________________________
dense_12 (Dense)            (None, 32)                2080

_________________________________________________________________
dense_13 (Dense)            (None, 32)                1056

_________________________________________________________________
```

6) Train and Test Validity



```
Text(0.5, 1.0, 'val accuracy')
```

7) PERFORMANCE ANALYSIS GRAPH



```
   Decision Tree       SVM  Random Forest  Neural Networks
0       0.976008  0.940499       0.993282         0.959693
```

8. CONCLUSION

A novel system for continuous user identification that we suggest is called KOLLECTOR Using cutting-edge learning techniques, we build a strong detector using sequential tapping data. The method still produces excellent accuracy when only three keystrokes are used, and it also provides more possibilities to make extra judgments that can lead to more accurate final decisions. KOLLECTOR is more adept at spotting fraudulent usage while still being extremely successful when compared to other shallow machine learning techniques. As a result, KOLLECTOR can be used in real life

9. REFERENCES

[1] T. Mogg, "Study indicates Americans misplaced cell phones last year valued at $30 billion," Mar. 2012. [Online]. Accessible at: http://www.digitaltrends.com/mobile/study-reveals-americans-lost-30-billion-of-mobile-phones-last-yearImpostor fraud: A cyber risk management dilemma, S. Watson, May 2015. [Online].Accessible at: treasuryandrisk.com/2015/05/05/

[2] Start learning your six-digit iPhone passcode, K. Knibbs, [3] Sep. 2015. [Online]. Accessible at http://gizmodo.com/start-memorizing-your-six-digit-iphone-passcode-1710072672.Most popular iPhone passcodes, D. Amitay, June 2011. [Online]. The most popular iPhone passwords are listed at http://danielamitay.com/blog/2011/6/13.

[3] Panda Security reports that "75 million US cellphones do not had their passwords put on." Sep. 2015. [Online]. You may access this information at: http://www.pandasecurity.com/mediacenter/tips/smartphone-risk-don't-use-password

[4] "Tapprints: Your finger taps have fingerprints," in Proceedings MobiSys 2012, ACM, Jun. 2012, pp. 323-336. E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury.

[5] "Accessory Password inference using accelerometers on smartphones," Proceedings of HotMobile 2012, ACM, February 2012. E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang.

[6] "TapLogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," Proceedings of WiSec 2012, ACM, April 2012, pp. 113–124.

[7] "Smudge attacks on smartphone touch screensx," Proceedings of WOOT 2010, August 2010. A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith.

[8]. Less is more: energy-efficient mobile sensing with thoughtless, by F. Ben Abdesslem, A. Phillips, and T. Henderson, in A novel spam short message categorization was described in Education Technology and Computer Science in 2009 by L. Duan, N. Li, and L. Huang. First International Workshop on ETCS'09. IEEE, 2009, vol. 2, pp. 168–171.

[9] "Personal identification based on iris texture analysis," IEEE transactions on pattern analysis and machine intelligence, vol. 25, no. 12, pp. 1519–1533, 2003. L. Ma, T. Tan, Y. Wang, and D. Zhang.

[10] Deep learning: techniques and applications, Foundations and Trends in Signal Processing, vol. 7, no. 3-4, pp. 197-387, 2014. [13] L. Deng, D. Yu, et al.

[11] Recurrent neural network regularisation, W. Zaremba, I. Sutskever, and O. Vinyals, arXiv preprint arXiv:1409.2329, 2014.

[12] A Convolutional Neural Network Approach for Objective Video Quality Assessment, IEEE Transactions on Neural Networks, P. Le Callet, C. Viard-Gaudin, and D. Barba, 2015.