CNN-BASED CONTINUOUS AUTHENTICATION OF SMARTPHONES USING MOBILE SENSORS

Sangam Man Buddhacharya*
Department of Electronics and Computer Engineering,
Pulchowk Campus, NEPAL
073bex438.sangam@pcampus.edu.np
Nishesh Awale
Department of Electronics and Computer Engineering,
Pulchowk Campus, NEPAL
073bex423.nishesh@pcampus.edu.np

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.09/Issue08/AUAE10083
Research Article | Open Access
Peer-review: Double-blind Peer-reviewed
Article ID: IJIRAE/RS/Vol.09/Issue08/AUAE10083
Received Date: 22, August 2022 | Accepted Date: 29, August 2022 | Available Online: 31, August 2022
Volume 2022 | Article ID AUAE10083 https://www.ijirae.com/volumes/Vol9/iss-08/37.AUAE10083.pdf
Article Citation: Sangam,Nishesh(2022). CNN-Based Continous authentication of Smartphone using Mobile Sensors. International Journal of Innovative Research in Advanced engineering(Vol.9,Issue8, pp.361–369).AM Publications,Ind
doi: https://doi.org/10.26562/ijirae.2022.v0908.37
BibTeX key sangam2022cnnbased
Academic Editor-Chief: Dr.A.Arul Lawrence Selvakumar

Abstract: With the advancement of technology, the smartphone has become a reliable source to store private details, personal photos, credentials, and confidential information. However, smartphones are easily stolen and it has become a suitable target for attackers. Usually, smartphones require only initial explicit authentication, once the initial login is passed all the information can be accessed easily. Hence, this paper proposes an efficient implicit, continuous authentication of the smartphone based on the user's behavioral characteristics. We propose architecture to differentiate legitimate smartphone owners from intruders. Our model relies on the smartphone's built-in sensors like an accelerometer, gyroscope, and GPS. The sensors respond according to the user's behavior which is recorded by the smartphone. We have used the rest filter model to separate motion data from rest since the rest data does not contain much information about the user's behavior. We have used Xgboost and Convolutional Neural Network as our rest filter and legitimate-intruder classifier respectively. Our system can predict legitimate and intruders in a few seconds. Our proposed CNN model has an achieved average accuracy of 95.79% in our custom dataset, which has further improved after integrating GPS data.
Keywords: Implicit continuous authentication, Convolutional Neural Network, Xgboost classifier, Accelerometer, Gyroscope

I. INTRODUCTION

Smartphones have become one of the most important parts of our daily lives. Nowadays people use smartphones not just for calling and texting but also for using social networks, doing online shopping or bank transaction, taking pictures, and storing private sensitive information [1][2]. When we go outside of our home, smartphone has become essential element as wallets and keys. With the increase of E-commerce and social networking, smartphones have become one of the most suitable targets for attackers to access personal and credential information. Current smartphones require the active participation of the user for explicit authentication, for example, entering password, face detection [3][4][5], fingerprint, etc. The problem with these types of authentication is that they do not re-authenticate the user once it passes the initial login. Therefore, the attackers have a free gateway to access the private information once the legitimate user login into the smartphone. The side channel attacker can luckily guess or brute force the password and can masquerade as a legitimate user to access proprietary or private data. To mitigate the problems of initial authentication, we require a continuous re-authentication system that recognizes the legitimate user throughout the process. Fingerprint, password, face detection [3][4][5], and iris scanning [6] require continuous participation from the user, so it would be annoying and interrupting for users. Therefore they are not suitable for continuous re-authentication. Considering this issue, we propose a reliable implicit continuous re-authentication system that doesn't require any participation of the user. Our method relies on the behavior of the user recorded by the smartphone's built-in sensors, for example, accelerometer, gyroscope, and GPS. The system continuously keeps track of the user without any interruption and prevents the intruder from accessing sensitive data and services.

Our system has different advantages in comparison with previous smartphone authentication methods: (1) Hassle free from typing the password frequently. A secure password is often not considered to be appropriate because they are too lengthy and contains alpha-numeric symbols so the user has to spend their precious time typing the password. (2) Leaves no hints for attackers to mimic the legitimate behavior as in the touchscreen pattern [7]. Touchscreen pattern always makes it easier for the side channel attacker to guess patterns since the user's fingertips often leave a distinguishing trace on the screen, which can indicate the pattern that was used to access the device. (3) Doesn't use any touchscreen information which might leak out sensitive information, for example, passwords, patterns, or PINs [8]. Our method is more suitable for E-commerce and other online transaction authentication purpose. It has an accuracy of up to 98.59% without using GPS data. When integrated with GPS, our method has shown improved performance. Using the authentication we can inject ad content to the appropriate user. If a legitimate user is using the phone, our system will detect the legitimate and pop relevant ad content to the legitimate users.

II. RELATED WORKS

Various approaches have been used for the authentication of users in smartphones. In the early days, authentication was done using secret passwords or tokens. If other people know the password, they can easily unlock the smartphone and access confidential data. Another approach for user authentication is biometric-based authentication. Biometrics such as face patterns, iris patterns, and fingerprint of a user is unique so they can be used for authentication. Niinuma et al. [4] and Hong et al. [5] show the use of face patterns for authentication.

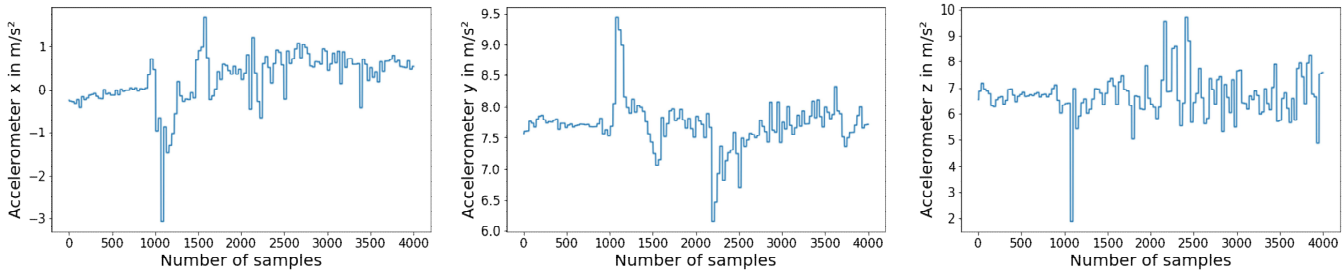


Fig. 1 Raw accelerometer data in x, y, and z directions from smartphone sensor

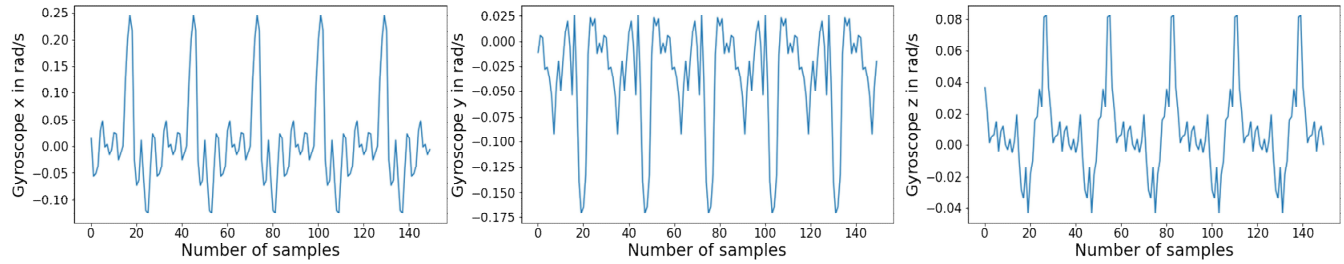


Fig. 2 Raw gyroscope data in x, y, and z directions from smartphone sensor

In the paper [5], face recognition and fingerprint verification are used which achieved an FAR of 1% and an FRR of 1.8% on the test set. Qi et al. [9] computed the user-specific iris features and used a genetic algorithm and Support Vector Machine (SVM) for feature selection and classification. Their approach achieved an accuracy of 99.92% (0% FAR and 0.08% FRR). Authentication systems related to breathing patterns have also been studied. Chauhan et al. [10] used three breathing patterns: sniff normal and deep breathing detected using the microphone of the smartphone. By using the Gaussian mixture model for the classifier, they reported an accuracy of 94%. Riva et al. [11] used a combination of face and voice recognition along with location signals for authentication. These biometric-based authentications require user participation and are only suitable for initial authentication.

To minimize user participation during authentication, different behavioral patterns such as gesture [12] [13], touchscreen [14], and gait [15] have been proposed. These approaches assume that every user has a unique way of picking up the smartphone during a call, keystroke on the touchscreen, and walking. In [16], HMOG (hand movement, orientation, grasp), tap, and keystroke dynamic features are used and the SVM model is trained for classification. Their method achieved an Equal Error Rate (ERR) of 7.16% in walking and 10.05% in sitting. Gesture-based method [17] uses accelerometer and gyroscope data with a dynamic time warping algorithm which achieved an accuracy of 96.3%. Trojahn et al. [14] developed a mixture of keystroke and handwriting-based authentication systems using the capacitive display with an FAR of 1% and FRR of 16%. Li et al. [18] monitored the user's finger movement patterns(sliding up, sliding down, sliding left, sliding right, tap gestures) and showed that the data of sliding up movement in portrait mode obtained the highest accuracy of 95.78%. In [15], the authors proposed a gait-based authentication using a k-nearest algorithm and reported an 8.24% Equal Error Rate (ERR). In addition, location information has also been used to authenticate the user. Data from GPS sensors, IP addresses of smartphones, and MAC (Media Access Control) have been used in the paper [19]. But the location data is sensitive and generally, users do not want to share their location information with a mobile app. Also, this approach requires explicit user participation.

Along with the smartphone sensors, data from different smart wearable devices have also been used for authentication. Cola et al. [20] exploited the accelerometer data from smart wearable devices and were able to do gait-based authentication. Their approach achieved an ERR in a controlled environment ranging from 5.7% (waist-mounted sensor) to 8.0% (trouser pocket). Lee et al. [21] proposed a context-based authentication system that used accelerometer and gyroscope data from smartphones as well as a smartwatch. Those data are used to first determine the context (motion or rest). Then, the Kernel Ridge Regression (KRR) classifier was used which achieved 98.1% authentication accuracy (0.9% FRR and 2.8% FAR).

In addition to traditional approaches, several deep learning approaches [22] have also been studied for continuous authentication. Amini et al. [23] used time and frequency domain features along with Long Short Term Memory (LSTM) on motion sensor data and reported 96.7% accuracy. Similarly, Zou et al. [24] used CNN and RNN in the gait data from motion sensors and achieved 93.7% accuracy. Centeno et al. [25] used an autoencoder model on accelerometer data under three scenarios: reading, writing, and browsing a map during sitting and walking. Their method achieved an ERR of 2.2%. These continuous authentication systems can detect legitimate users from an intruder without user participation using behavioral data from different sensors.

III. METHODS

In this section, we describe our proposed architecture along with the detailed procedure for the data collection, data pre-processing, rest filter model, kernel size, and window size selection.

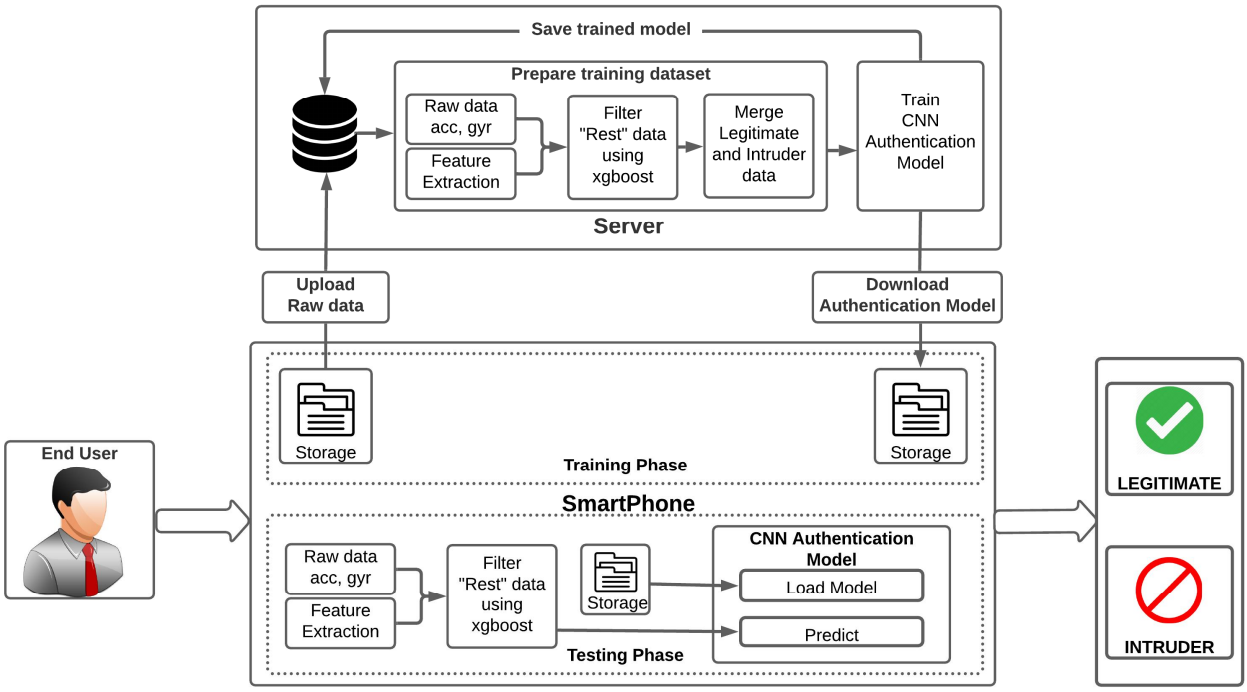


Fig. 3 Our proposed architecture

A. Architecture Overview

Our proposed architecture is shown in Fig. 3. It contains 4 main components: Server, Smartphone, Rest filter model, and CNN authentication model. There are two different phases: training and testing, which works separately.

1) Training Phase: Initially, our model is new to the user so it behaves randomly. Therefore, it must be trained with the user's data. During the training phase, our system continuously monitors and collects sensor (accelerometer and gyroscope) data from the smartphone. Our main aim is to find the pattern of the users when they use their phone in the usual way while performing the activities like typing, scrolling, calling, chatting, walking, holding, etc. so our system collects the data only when the user is active. The collected data are temporarily stored in local storage until the data are sufficient for training the model. In our experiment, when stored data reaches up to 5 MB, it is automatically uploaded to the server. The smartphone also sends GPS location (latitude, longitude) information during uploading to track the user's location. To protect users' privacy, each user is identified with their unique android id. In the server, when the raw data is uploaded, it is retrieved from the database and is pre-processed by the rest filter model. We have considered the phone resting in the table as "rest data". Rest data is invalid for training since it has no features to distinguish legitimate from intruders, so we remove it. To train the model, it requires legitimate and intruder data. Therefore, the training dataset is prepared by merging the user's data as legitimate and other participating users' data as intruders. We select 5 - 20 intruders randomly such that the legitimate and total intruders have an equal number of data. The number of data from each intruder depends upon the number of the selected intruders and legitimate data.

If the number of intruders is less then, the data from each intruder is more and vice versa. The CNN authentication model is trained with the prepared dataset. After training, the models are saved to the database and downloaded to the smartphone. The behavior of the legitimate user might change according to the time so; we record the latest data weekly and train our pre-trained model. The newly collected data are kept at the top of the stack while the oldest data are removed from the bottom of the stack.

2) Testing Phase: Once the model is trained, the smartphone is ready for continuous authentication. In the testing phase, the sensor's data is continuously accumulated from 6 different channels accelerometer (x, y, and z) and gyroscope (x, y, and z) at 50 Hz. A time-series data from different channels are segmented with a window of size 200x6. Features are extracted from segmented data and are fed to the rest filter model. Based on the features extracted, the rest filter model identifies the invalid/rest data. Only valid data is further processed. Finally, the CNN authentication model classifies whether the input data is legitimate or intruder. To achieve promising results, we have added some constraints. The users are classified as legitimate or intruders only if the model consistently outputs the same class (legitimate or intruder) thrice. For the authentication task, it is required to have a minimum false positive rate so we plot the area under the curve (AUC) for each user and find the optimal threshold. We have explicitly used the advantage of GPS location to track the legitimate user. As shown in Fig. 4, a circle of radius x km (threshold), with the center as the user's recorded (during training) GPS location (latitude, longitude) is created where the radius is the threshold that needs to be considered. All the locations during the training phase are recorded as valid user locations.

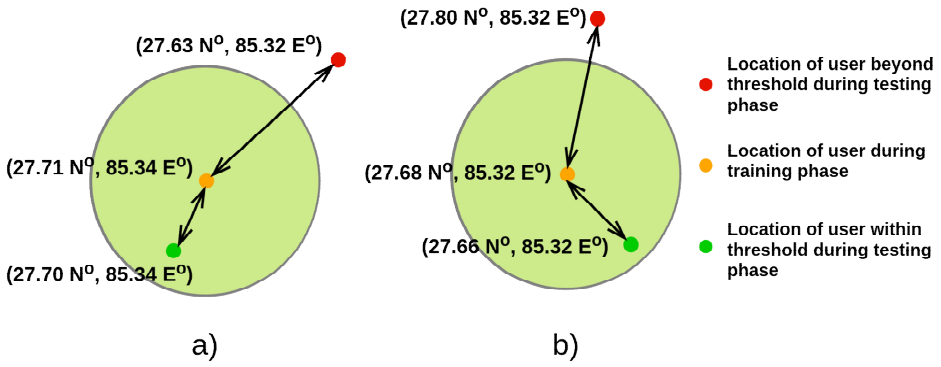


Fig. 4 Enforcing the penalty to its predicted probability, according to the current GPS location. "a" and "b" are the 1st and 2nd locations of the user during training.

During testing, if the current GPS location of the user lies within a circle, we consider it valid and don't give any penalty to its predicted probability else we give a penalty according to the threshold. Enforcing the penalty improves the robustness of our authentication model since; to overcome the penalty the user needs to have a higher prediction probability greater than the optimal threshold. (I.e. For 0.8 optimal threshold 0.85 penalty is given so the users need to have more than 0.95 prediction probability where, 0.95 (prediction probability) $*$ 0.85(penalty) = 0.8075 > 0.8 (threshold criteria for classified output to be valid.)).

B. Data collection

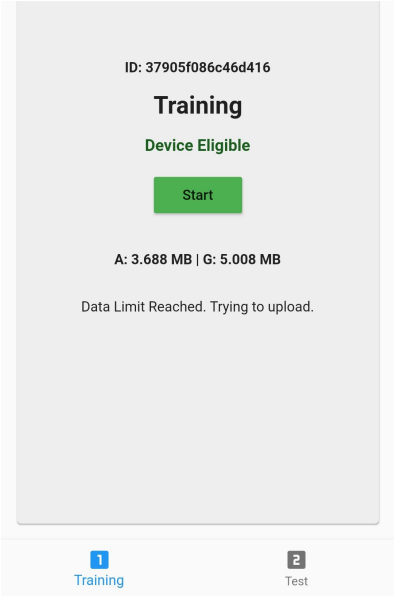


Fig. 5 Mobile app for collecting sensor data

Our CNN authentication model is based on a supervised machine learning approach. For training the model, a large number of legitimate and intruder data are required. So, we built a sensor data collection app, as shown in Fig. 5. Smartphone has a number of sensors like a touchscreen, proximity, magnetometer, accelerometer, gyroscope, GPS, etc. As inspired by Lee et al. [21], we have selected an accelerometer, gyroscope, and GPS sensor as major components for authentication since they are independent of environmental factors. Human activities (typing, scrolling, calling, walking, holding, etc.) often have a frequency of less than 20Hz. So, the data were collected at the rate of 50Hz to avoid aliasing. To carry out the experiment, we distributed our app to 15 different users and each user is distinguished with their android-id. During a single upload, it collects 5MB of each sensor's data. On average, each user used their phone for about four hours a day. The users were not explicitly taught or forced to do any activities. They used their phone in the usual way. During the data collection activities like typing, scrolling, calling, walking, holding, chatting, etc. were frequently performed. Our app runs in the background and collects the data only when the user passes the initial security login.

C. Filtering the rest data

TABLE I - Rest/motion classification with different machine learning algorithms. xgboost outperformed all the other Classification models, for the same set of training and testing data. The best result is shown in bold.

Model	Accuracy (%)	Support (Rest)	Support (Motion)
Logistic Regression	95	600	600
SVM	95	600	600
Random Forest	97	600	600
Xgboost	98	600	600

As the raw sensor data from the smartphone consists of only accelerometer and gyroscope values in three directions, they are not good enough for feeding directly to train the machine learning model. So, the raw data were grouped into window size of 200, and features from time, as well as a frequency domain, was extracted using the tsfel library. We found that the following were the dominant features for our dataset based on the value of mutual information. 1. Absolute energy 2. Area under the curve 3. Interquartile range 4. Kurtosis 5. Maximum value 6. Minimum value 7. Mean 8. Median absolute deviation 9. Standard deviation 10. Maximum frequency 11. Entropy 12. Negative turning points 13. ECDF (Empirical Cumulative Distribution Function) Percentile along the time axis 14. Median absolute difference 15. Spectral distance 16. Wavelet energy 17. Wavelet variance 18. Power spectrum density bandwidth. The corresponding labels were computed by taking the mode of the values within the same window of 200 rows. After computing the features, the missing values and invalid numbers were replaced by the mean of the corresponding feature. The highly correlated features with Pearson correlation coefficient value greater than 0.95 were also removed. Only one of such features was kept. Also, the features with zero variance were removed. After selecting the required features, the data were normalized to have zero mean and unit variance so that all the features come to the same scale.

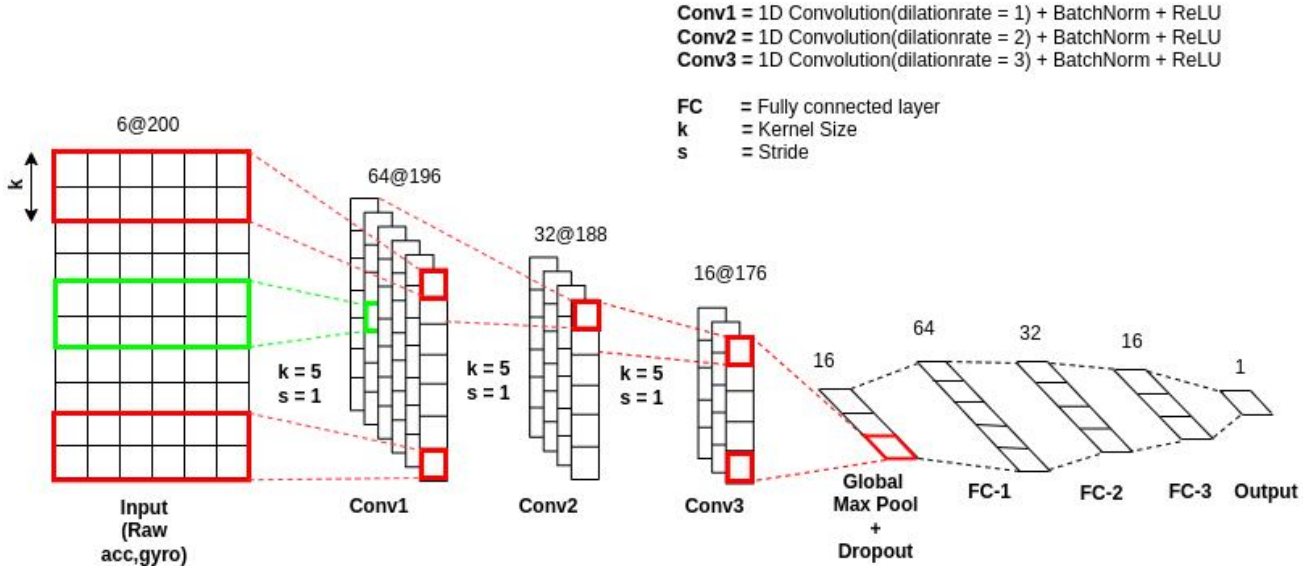
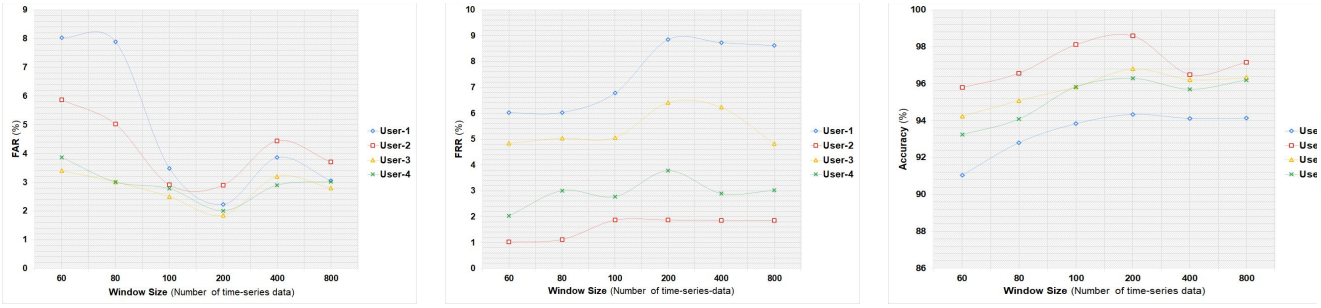


Fig. 6 CNN authentication model architecture

We experimented with different machine learning models like Logistic Regression, Random Forest, Support Vector Machine (SVM), and XGBoost. While comparing the result of K-fold cross-validation of different models, we found that the XGBoost model performed best in our case. As shown in Table I, the XGBoost model gave 98% accuracy on our dataset. Since the rest data consists mostly of nearly constant values, detecting rest data was easy for the XGBoost model

D. CNN user authentication model

Our proposed architecture for the authentication model can be seen in Fig. 6. For authentication, continuously sampled time series data is collected. We have used 1D convolution over the time series data from different channels (acceleration-X, acceleration-Y, acceleration-Z, gyroscope-X, gyroscope-Y, and gyroscope-Z). Our model consists of 3 convolutional blocks, a global max-pooling layer, dropout, and fully connected layers. Each convolutional block consists of 1D convolutional operation, batch normalization, and ReLU activation function. To increase the receptive field and avoid signal decimation, we have replaced the max-pooling layer with atrous convolution. The grinding effect exists if the entire block uses the same dilation rate [26]. To remove the grinding effect, we have used a dilation rate of 1, 2, and 3 respectively for each convolution. The three convolutional blocks extract abstract representations of the input time-series data.



(a) (b) (c)

Fig. 7 (a), (b), and (c) are FAR, FRR and accuracy respectively with different window sizes for 4 different users. Considering the graphs, the optimal size of the window is 200. As our sampling frequency is 50 Hz, the window time is 4 seconds.

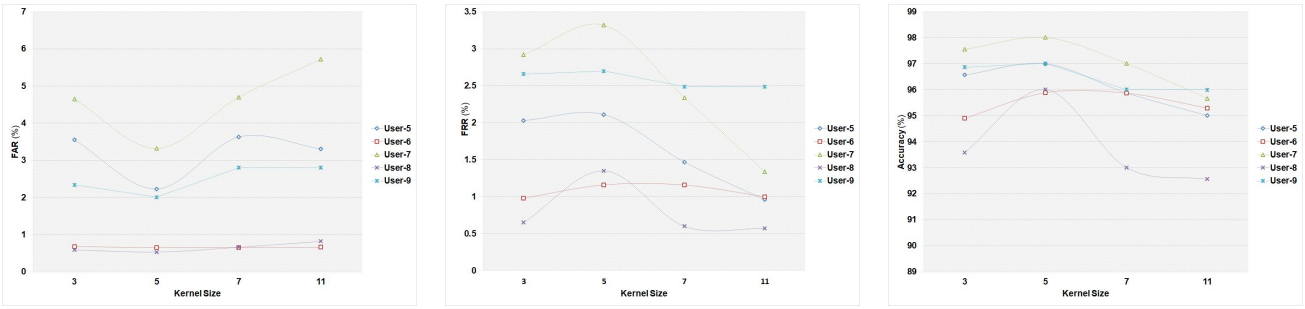


Fig. 8 (a), (b), and (c) are FAR, FRR and Accuracy respectively with a window of 200 for the different kernel sizes. For most of the users, the FAR, FRR, and accuracy are optimal when the kernel size is equal to 5. To avoid bias towards the users, we have experimented with different users for kernel size selection.

E. Selection of Window and kernel size

We have performed an experiment with four different users to select an appropriate window size for our CNN authentication model. We have selected the optimal result from FAR, FRR, and accuracy graphs. From Fig. 7, we found window size 200 to be more optimal with lower FAR, higher FRR, and highest accuracy. As our sampling frequency is 50 Hz, the window is of 4 seconds which is appropriate for real-time application. The results of our CNN authentication model were calculated on the data of fifteen users. Also, the FAR and FRR were calculated for the users and the graph has been shown in Fig. 8. From the graph, we can see that the FRR is lowest for the user whose FAR is the highest and vice-versa. Also, the optimal size of the kernel from the graph of FAR and FRR was found to be 5. So, we calculated the accuracy for fifteen users using the kernel size of 5, and the results are shown in Table II.

IV. RESULTS

For the authentication task, we compared the result of both machine learning as well as deep learning-based methods. As shown in Fig. 9, Convolutional Neural Network (CNN) has the highest accuracy when compared to the result of traditional machine learning-based methods. Also, as the size of the dataset was increased, the CNN model outperformed the traditional machine learning models by a large margin. So, we choose the CNN model for authentication which obtained an average accuracy of 95.79% on our dataset. The accuracy of our proposed model for different users is shown in Table II. We evaluated the results using only accelerometer and gyroscope data. While calculating the results, all the data in our server for the user were used. Due to this, the total data for each user is different which can be seen in the Support (Legitimate) column of Table II. The intruder data for a particular user was made by merging the data from all the other users. From Table II, we can see that the accuracy of our CNN authentication model is 91.67% (lowest value) for User-14 and 98.59% (highest value) for User-2.

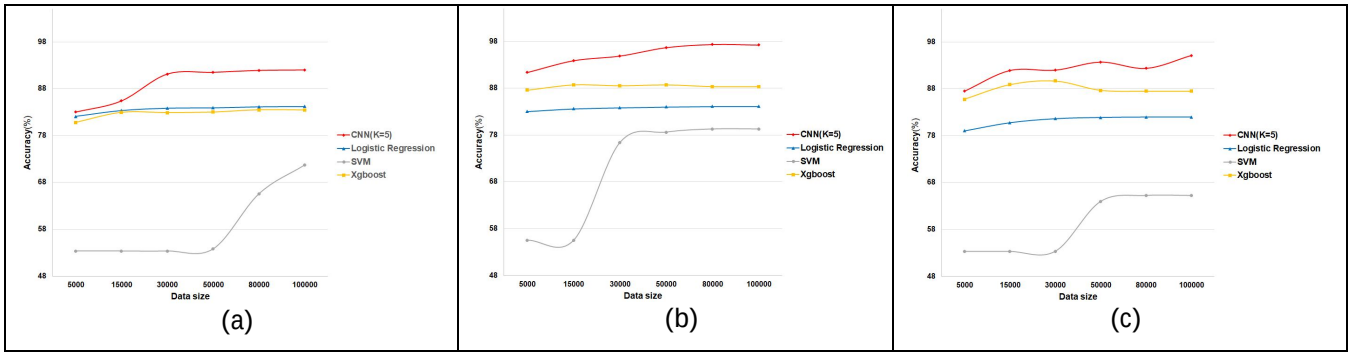


Fig. 9 (a), (b), and (c) compare the accuracy of the CNN model and the traditional machine learning models (Linear Regression, SVM, and Xgboost) on different dataset sizes for 4 different Users.

TABLE II - Accuracy of our proposed model (kernel = 5, window size = 200) for different users. User-14 has a lower accuracy in comparison to other users. Because during the training phase, intruders might have used legitimate phones.

Users	Accuracy (%)	Support (Intruder)	Support (Legitimate)
User-1	94.34	9945	9852
User-2	98.59	21683	21691
User-3	96.80	17231	16933
User-4	96.29	20792	21004
User-5	97.00	34232	33647
User-6	95.89	32319	32420
User-7	98.01	34595	34405
User-8	96.01	24595	24508
User-9	97.02	12045	12050
User-10	92.45	25001	25015
User-11	93.89	19880	19901
User-12	96.67	31020	31032
User-13	94.35	10112	10124
User-14	91.67	13450	13480
User-15	97.89	34400	34405

V. CONCLUSION

In this paper, we proposed architecture for continuous re-authentication in the smartphone to protect private and confidential information. For training our models, we collected the data from 15 different users with an android app at a rate of 50 Hz. For the rest data filter model, we tested with the different machine learning algorithms and found the XGBoost model as the best with an accuracy of 98%. To prepare the dataset for training the rest filter model, we selected the dominant features based on the mutual information. We trained a convolutional neural network separately for each user on the rest filtered data and found average accuracy of 95.79%. We also performed experiments with different kernel sizes and window sizes. The kernel size of 5 and window size of 200 was found to be optimal for our purpose. Comparing our approach with the traditional machine learning models (i.e. SVM, Logistic regression, and Xgboost), we found our model to be more accurate with increasing data. We also observed that the performance has increased after integrating the GPS data with our prediction model. Our method can predict legitimate and intruders in a few seconds so it can be used for real-time application. We have further plan to embed our method in E-commerce and other online transaction authentication purpose. Ads injection to the relevant legitimate user can be the rising application of our proposed method.

REFERENCES

1. Youngho Kim, Tae Oh, and Jeongnyeo Kim. “Analyzing User Awareness of Privacy Data Leak in Mobile Applications”. In: Mobile Information Systems 2015 (2015), pp. 1–12.doi: 10.1155/2015/369489. url: https://doi.org/10.1155/2015/369489
2. J. P. Achara et al. “Mobilitics: Analyzing Privacy Leaks in Smartphones”. In: ERCIM News 2013 (2013).
3. Kai Xi, Jiankun Hu, and Fengling Han. “Mobile device access control: an improved correlation based face authentication scheme and its Java ME application”. In: Concurrency and Computation: Practice and Experience 24.10 (July 2011), pp. 1066–1085. https://doi.org/10.1002/cpe.1797

4. K. Niinuma, U. Park, and A. K. Jain. “Soft Biometric Traits for Continuous User Authentication”. In: IEEE Transactions on Information Forensics and Security 5.4 (2010), pp. 771–780. https://doi.org/10.1109/tifs.2010.2075927
5. Lin Hong and Anil Jain. “Integrating faces and fingerprints for personal identification”. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 20.12 (1998), pp. 1295–1307. https://doi.org/10.1109/34.735803
6. M. Qi et al. “User-Specific Iris Authentication Based on Feature Selection”. In: 2008 International Conference on Computer Science and Software Engineering. Vol. 1. 2008, pp. 1040–1043. https://doi.org/10.1109/csse.2008.1060
7. Adam J. Aviv et al. “Smudge Attacks on Smartphone Touch Screens”. In: Proceedings of the 4th USENIX Conference on Offensive Technologies. WOOT’10. Washington, DC: USENIX Association, 2010, pp. 1–7.
8. Zhi Xu, Kun Bai, and Sencun Zhu. “TapLogger”. In: Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks - WISEC ’12. ACM Press, 2012. https://doi.org/10.1145/2185448.2185465.
9. M. Qi et al. “User-Specific Iris Authentication Based on Feature Selection”. In: 2008 International Conference on Computer Science and Software Engineering. Vol. 1. 2008, pp. 1040–1043. https://doi.org/10.1109/csse.2008.1060
10. Jagmohan Chauhan et al. “BreathPrint: Breathing Acoustics-Based User Authentication”. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. MobiSys ’17. Niagara Falls, New York, USA: Association for Computing Machinery, 2017, pp. 278–291. isbn: 9781450349284. https://doi.org/10.1145/3081333.3081355
11. Oriana Riva et al. “Progressive Authentication: Deciding When to Authenticate on Mobile Phones”. In: 21st USENIX Security Symposium (USENIX Security 12). Bellevue, WA: USENIX Association, Aug. 2012, pp. 301–316. https://doi.org/10.1109/ms.2011.67
12. Mohammed A. Alqarni et al. “Identifying smartphone users based on how they interact with their phones”. In: Human-centric Computing and Information Sciences 10 (2020), pp. 1–14. https://doi.org/10.1186/s13673-020-0212-7
13. Mauro Conti, Irina Zachia-Zlatea, and Bruno Crispo. “Mind How You Answer Me! Transparently Authenticating the User of a Smartphone When Answering or Placing a Call”. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. ASIACCS ’11. Hong Kong, China: Association for Computing Machinery, 2011, pp. 249–259. https://doi.org/10.1145/1966913.1966945
14. M. Trojahn and F. Ortmeier. “Toward Mobile Authentication with Keystroke Dynamics on Mobile Phones and Tablets”. In: 2013 27th International Conference on Advanced Information Networking and Applications Workshops. 2013, pp. 697702. https://doi.org/10.1109/waina.2013.36
15. C. Nickel, T. Wirtl, and C. Busch. “Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm”. In: 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. 2012, pp. 16–20. https://doi.org/10.1109/iih-msp.2012.11
16. Z. Sitová et al. “HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users”. In: IEEE Transactions on Information Forensics and Security 11.5 (2016), pp. 877–892. https://doi.org/10.1109/tifs.2015.2506542
17. Lee, Wei-Han et al. “Secure Pick Up: Implicit Authentication When You Start Using the Smartphone.” Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies (2017). https://doi.org/10.1145/3078861.3078870
18. Lingjun Li, X. Zhao, and G. Xue. “Unobservable Re-authentication for Smartphones”. In: NDSS. 2013.
19. F. Zhang, A. Kondoro, and S. Muftic. “Location-Based Authentication and Authorization Using Smart Phones”. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. 2012, pp. 1285–1292. https://doi.org/10.1109/trustcom.2012.198
20. G. Cola et al. “An unsupervised approach for gait-based authentication”. In: 2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN). 2015, pp. 1–6. https://doi.org/10.1109/bsn.2015.7299423
21. W. -H. Lee and R. B. Lee, "Implicit Smartphone User Authentication with Sensors and Contextual Machine Learning," 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017, pp. 297-308, https://doi.org/10.1109/dsn.2017.24
22. Hasan Can Volaka et al. “Towards Continuous Authentication on Mobile Phones using Deep Learning Models”. In: Procedia Computer Science 155 (2019). The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019),The 14th International Conference on Future Networks and Communications (FNC-2019),The 9th International Conference on Sustainable Energy Information Technology, pp. 177–184. doi: https://doi.org/10.1016/j.procs.2019.08.027
23. Sara Amini et al. “DeepAuth: A Framework for Continuous User Re-Authentication in Mobile Apps”. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. CIKM’18. Torino, Italy: Association for Computing Machinery, 2018, pp. 2027–2035. isbn: 9781450360142. doi: 10.1145/3269206.3272034. url: https://doi.org/10.1145/3269206.3272034.

24. Q. Zou et al. “Deep Learning-Based Gait Recognition Using Smartphones in the Wild”. In: IEEE Transactions on Information Forensics and Security 15 (2020), pp. 3197–3212. https://doi.org/10.1109/tifs.2020.2985628
25. M. P. Centeno, A. v. Moorsel, and S. Castruccio. “Smartphone Continuous Authentication Using Deep Learning Autoencoders”. In: 2017 15th Annual Conference on Privacy, Security and Trust (PST). 2017, pp. 147–1478. https://doi.org/10.1109/pst.2017.00026
26. P. Wang et al., "Understanding Convolution for Semantic Segmentation," 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 1451-1460, https://doi.org/10.1109/wacv.2018.00163
27. M. Barandas et al. TSFEL: Time Series Feature Extraction Library. SoftwareX 11, 100456, 2020. https://doi.org/10.1016/j.softx.2020.100456
28. M. Abuhamad, A. Abusnaina, D. Nyang and D. Mohaisen, "Sensor-Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey," in IEEE Internet of Things Journal, vol. 8, no. 1, pp. 65-84, 1 Jan.1, 2021, https://doi.org/10.1109/jiot.2020.3020076
29. A. Alzubaidi and J. Kalita, "Authentication of Smartphone Users Using Behavioral Biometrics," in IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 1998-2026, thirdquarter 2016, https://doi.org/10.1109/comst.2016.2537748
30. C. Shen et al. “Performance Analysis of Motion-Sensor Behavior for User Authentication on Smartphones”. In: Sensors. 2016. doi: 10.3390/s16030345 url: https://doi.org/10.3390/s16030345.
31. C. Shen, Y. Li, Y. Chen, X. Guan and R. A. Maxion, "Performance Analysis of Multi-Motion Sensor Behavior for Active Smartphone Authentication," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 1, pp. 48-62, Jan. 2018, https://doi.org/10.1109/tifs.2017.2737969
32. M. Abuhamad, T. Abuhmed, D. Mohaisen and D. Nyang, "AUToSen: Deep-Learning-Based Implicit Continuous Authentication Using Smartphone Sensors," in IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5008-5020, June 2020, https://doi.org/10.1109/jiot.2020.2975779
33. Ö. D. Incel et al., "DAKOTA: Sensor and Touch Screen-Based Continuous Authentication on a Mobile Banking Application," in IEEE Access, vol. 9, pp. 38943-38960, 2021, https://doi.org/10.1109/access.2021.3063424

