International Journal of Advances in
Soft Computing and Intelligent Systems (IJASCIS)
2022, Vol 01, Issue 02, 62-71

Science Transactions © 2022

Original Paper

EXPLORING RAW SWIPE VECTORS FOR CONTINUOUS AUTHENTICATION OF SMARTPHONE USERS

Rupanka Bhuyan[1,3], and S Pradeep Kumar Kenny[2]
[1]Department of Computer Science, St. Joseph University, Dimapur, Nagaland, INDIA
rupanka@yahoo.com
[2]Department of Computer Science, St. Joseph University, Dimapur, Nagaland, INDIA
kenny_isles@yahoo.co.in
[3]ICFAI University Nagaland, Chumoükedima, Nagaland, INDIA

ABSTRACT

Touch based Continuous Authentication (TCA) is a security method that is useful for perpetually validating a user's identity. In the setting of touchscreen based smartphones, one type of TCA uses the swipe characteristics of the user to unobtrusively extract hidden patterns to ascertain his or her identity. However, most of the swipe based TCA methods require this raw touch input data to be pre-processed and scaled to a different form to be of any use for accurate prediction. Further, most of these methods require a user to input multiple swipes before arriving at the authentication decision. This work explores the applicability of methods on the raw swipe data and also attempts to attest users with minimum number of swiping inputs to arrive at the authentication decision at the earliest to minimize damages in case of any unauthorized access. Decent authentication performance is achieved with un-processed and minimal swipe inputs from the user.

KEYWORDS

Continuous Authentication (CA), Touch based CA, Swipe-Vectors, Extremely Randomized Trees.

1. Introduction

One of the most popular devices in the current times is the smartphone. Owing to its portability and computing capability almost equal to those of desktops and laptops, these devices have managed to capture the consumers' interest at an unprecedented level. Predicted market share for smartphones for 2022 is 1.646 billion units [1]. Smartphones carry a touchscreen for user interaction which is an efficient way for incorporating both input and output interfaces into one [2] making it very convenient for the user's interaction.

This in turn influences smartphone users to keep most of their data including important ones in the device. This may include details of passwords, online shopping credentials and confidential files and so on. Unlike a desktop PC or laptop used in secure premises the smartphone is used anywhere. This justifies the need for additional security for the device.

One type of security mechanism is the initial login procedure by the device's genuine user or owner which requires the incumbent's identity to be verified once to unlock it. This is generally known as One Time Authentication (OTA) which include methods like face scan, biometrics, passwords and personal identification numbers (PINs).

The other type of security mechanism is where continuous verification of the user's identity is carried out while the user is using the device after the OTA procedure is completed. Continuous Authentication (CA) is the term used for this mechanism [3] which usually runs as a recurrent background activity on the device being used.

The rest of the paper is organized as follows. Section 2 provides the background concepts related to Touch based Authentication (TCA). Section 3 introduces the concepts pertaining to Swipes based and Unobtrusive TCA. Section 4 discusses various contemporary works related to TCA. Sections 5 and 6, states the contribution of the paper and introduces the dataset. Section 7 illustrates the various methods used. In section 8, the performances of the methods are discussed. Section 9 contains the Results and Discussions. Section 10 concludes the paper.

2. Touch based Continuous Authentication and its Significance

By definition, Touch based Continuous Authentication (TCA) is a subset of CA wherein user's genuineness is determined by the extracting identifiable behavior from the users' touch activities on the touchscreen of the device. Touch activities include tapping, swiping, typing on the keypad to name a few [4]. Given the fact that smartphones are built with a touchscreen, TCA are an appropriate means for implementing CA in the device.

As Stylios [5] indicated, a user's touch behavior on the touchscreen of the device is unique and a specific digital signature can be created from it. This can be done by extracting subtle features from the touch behavior pattern which are generally not visibly detectable by the user.

The importance of TCA cannot be ignored in view of the additional security need-ed for these devices as mentioned in section 1. After initial login (OTA) by the user wherein the device is left unattended, there may come up a situation where an intruder attempts to access it. There would be no way of detecting this breach unless TCA is active in the device.

Other than being a secondary line of defense, TCA has some additional ad-vantages. TCA works without any requirement of any special or additional hardware. Data required for implementing TCA can be generated from the usual user interactions and inputs which are readily available. Digital profiles based on TCA can be generated as an when any required either from scratch or in situations where an existing profile is compromised [6].

A desirable trait for any TCA is not only its detection accuracy but its detection speed. More elaborately, if an intruder is using the smartphone, then the TCA should be proactively notice the intruder before several touch operations are performed [7]. The detection should be done within a few touches.

3. Swipes based Unobtrusive Continuous Authentication

The granularity of the TCA paradigm can be further refined into a more detailed level introduced as Swipes based Unobtrusive Continuous Authentication (SUCA).

Firstly, Unobtrusive CA can be defined as the process of continuous authentication where the user's attention or input is not explicitly required. With the normal inputs and actions executed by the user over the touchscreen's surface, the CA can be performed, hence the name unobtrusive.

Secondly, the term swipe implies a touch activity on the touchscreen by the user to get a certain result which may include - scroll up, input a pattern, zoom, etc. As observed by Neal et. al. [8], it is the commonest activity of a user using a smartphone with a touchscreen. Technically a swipe is represented by a tuple $(p_{i,\ldots})$ where $1 \leq i \leq n$ and p_i represent a point on the touchscreen's surface touched by the user [9]. Usually, consecutive p_is are in close proximity to each other. Thirdly, for each tuple there may be additional parameters such as pressure, area, velocity, acceleration, to name a few. A swipe will be denoted by the word used 'swipe-vector' in this text hereafter.

4. Related Works

This work particularly focuses on the swipe inputs called in this text as swipe-vectors although there are a quantity of works where data from sensors like magnetometers, accelerometers and gyroscopes have been used in addition to these swipe inputs. However, as per Neal's observation [8], noise is easily introduced into the data stream due to use of these additional sensors because of which this work. Another reason for this is the undeniable fact that availability of all the other types of sensor data may not be available for all types of smartphones whereas the swipe-vector data is commonly available.

It is noticed that the entire gamut of research works in the subdomain in question revolves around a few areas of focus. Hence, the discussions of all these research works are separated into groups containing (i) Neural Network (NN) based methods, (ii) Decision Tree (DT) based methods, (iii) Support Vector Machines based methods, and (iv) others, as given below.

Using a Siamese Neural Network, Acien [10] on a dataset containing drag and drop swipe-vectors, 87% accuracy could be reached. Long Short Term Memory (LSTM) neural networks were used in two works carried out independently respectively by researchers Hochreiter et. al. [11] and Liu et. al. [12] resulting in performance accuracies of 73.10% and 87.72%. A Convolutional Neural Network (CNN) was employed in Debard's works [13] to a dataset of 27 users with 6591 touch movements. Debard could reach an accuracy of 89.96%. A One-class Random Maxout Probabilistic Network was used by Choi [14] on the Touchalytics [4] and HMOG [15] datasets separately; the identity a genuine user could be verified after a series of 11 continuous touch strokes. Lin's work [1] using Back Propagation Neural Network (BPNN) and Radial Basis Function Network (RBFN) along with other methods achieved accuracy hits up to 79%. In the works of Samet [16], Multi-Layer Perceptron (MLP) was incorporated to authenticate users based on their swiping and typing activities. Based on a dataset of 8 users, authentication accuracies of 96 to 100% were reported.

An important fact to be considered with respect to any neural network based methods is that the data should be pre-processed ahead of being fed in to the network; in other words, data scaling using (i) standardization [17] or (ii) normalization [17]. This can be carried out based on the concept that the entire dataset is available beforehand.

Researches implementing some form of the Support Vector Machines (SVM) are discussed in this paragraph. Lin [1] mentioned above also used SVM in the said work. Tornai used a type of SVM in a 13 user dataset in authenticating users; accuracies obtained were up to 91% [18]. With a minimum of 9 touch operations, Yang's work [19] could correctly differentiate the identity of a genuine user from that of an impostor; an accuracy of 95.85% was clocked in the 45 user dataset using a One Class Support Vector Machine (OcSVM). Sharma et. al. used a SVM based ensemble to continuously determine the proper identity of the users of an app and demonstrated accuracy scores up to 93% [20]. In his work, Barlas [21] also implemented a One Class SVM on a 30 user dataset; authentication accuracies reached a demonstrated maximum value of 79%. In the Support Vector Machine based technique proposed by Zhang authentication accuracy score was clocked at 97% along with F1 score of 93.7%, respectively. This was demonstrated in a dataset of 10 users [22]. A disadvantage with SVM based methods is that a number of parameter settings are to be tweaked for an SVM to yield good accuracies.

Related works involving Decision Tree based methods are discussed this point onward. Syed et. al. work demonstrated that an user can be authenticated with the inputting of at least 5 strokes by the user. With accuracies unspecified, the method based on Random Forests (RF) used 19373 strokes each from 31 users [23]. Yet another Random Forest based technique was demonstrated whereas in Gunn's work [24] a dataset with 5 users was used to achieve authentication accuracy score of 99.0361%. Meanwhile, Leingang et. al. used the tree based J48 technique on the 100 user HMOG dataset obtaining accuracies of 88.69% [25]. J48 was also used in Lin's works [1]. J48, Random Committee and Random Forest algorithms were also used

in the earlier mentioned works of Samet [16]. An important property of Decision Tree based methods are that they are robust in handling data of varying scales.

Other types of methods are discussed henceforth. Both Naïve Bayes and Bayes Net based techniques were implemented in the earlier mentioned works by Samet [16] on an 8 user dataset. Shankar achieved 95 percent and 97 percent authentication accuracies in authenticating users while they were performing activities in the walking and sitting states respectively; this was exhibited in a dataset of 10 users using a Deep Auto Encoder and Softmax Regression technique [26]. Pre-processing of data (Data normalization, Scaling) was carried prior to implementing these techniques.

5. Contributions of this paper

This paper explores methods for SUCA of smart phone users that:

─ performs authentication on the basis of unprocessed or primitive swipe-vectors reducing processing overhead.

─ consumes the least user input therefore resulting in faster authentication.

6. The Dataset

Datasets pertaining to swipe characteristics for continuous authentication are few in numbers. Moreover, not all the datasets include all type of (swipe-able touchscreen) devices, particularly both smartphones and tablets. This work uses a dataset contributed by Bellman [27]. It contains 306096 parameter values extracted from 10932 swipe-vectors. These vectors were in turn collected from 117 smartphone users.

7. The Methods

Five methods are identified based on their suitability to handle the raw or un-processed touch based swipe characteristics data of the users. The fundamental theories pertaining to each of these methods are discussed in the subsections subsequent this section.

7.1. Decision Trees (DT)

With applicability to both classification as well as regression problems, Decision Trees (DT) are used for predicting the class or value of an input variable. A Decision Tree learns its decision rules from inputted training data with class labels. In composition a Decision Tree looks like an inverted tree made up of connected nodes wherein at the top of the tree there is a root node. Normally, there are branches from this root node which progress downward to connect more nodes. The number of nodes towards the downward direction of the Decision Tree increases owing to the increase of branches.

In situations where the class label of a variable is not known, a route is drawn from the root to a leaf node which contains that variable's class prediction. While constructing the Decision Tree, two metrics are calculated, namely (i) Entropy, and (ii) Information Gain respectively. The definitions of these are furnished below.

Entropy can be best defined as a measure of homogeneousness or similarity or contradictorily the degree of impurity in context of a given dataset. Considering a dataset D comprising of both positive and negative samples with respect to certain rules, the Entropy [28] of is derived as depicted below in equation (1) whereas the positive and negative instance portions of D_{RAW} are depicted by p_{pos} and p_{neg} respectively.

$$\text{Entropy}\ (D_{RAW}) = -\left(p_{pos} \text{Log}_2 p_{pos} + p_{neg} \text{Log}_2 p_{neg}\right) \quad (1)$$

Since categorization of the training set is based on a specific attribute or parameter at a given step of building the Decision Tree, the efficiency of this categorization needs to be measured. This is where the metric Information Gain (IG) comes into play. Ideally, IG is defined as the predicted reduction in Entropy obtained by splitting the dataset based on the given attribute. Considering a collection of datasets D_{COL}, equation (2) below [29] illustrates the information gain IG(S,P) of a parameter P in relation to D_{COL} where Values(P) is the set of all possible values of attribute P, D_{COL}^{V} is the subset of D_{COL} for which attribute P has value v.

$$IG (D_{COL} , P) = E (D_{COL}) - \sum_{v \in Values(P)} \{ | D_{COL}^{v} | / | D_{COL} | \} \quad (2)$$

7.2 Random Forests (RF)

In essence, the Random Forest (RF) is made up of multiple tree-structured classifiers which can be depicted as $\{h(x,\theta_k),k = 1,...\}$ where the $\{\theta_k\}$ are random vectors which are independent and identically distributed. Given input x, each one of these tree classifiers yields a unique vote for the most prominent class [30].

The primary working principle of the RF method is furnished below:

(i) From the original set of (say, k) records, n random records are drawn out

(ii) For each of the n samples obtained in step (i), individual and independent Decision Trees (DT) are created

(iii) There is an output from each of the DTs produces in step (ii)

(iv) The final output is evaluated based on majority voting

Points considered with regard to Random Forests that have been considered noteworthy for this work are:

– While building each of these individual and independent trees, all the features or parameters from the original dataset are not used. This in turn results in a relatively smaller dataset which translates to faster processing,

– Moreover, being independent, each of these trees can be created in parallel order which also adds up to faster processing speeds.

7.3 Extreme Gradient Boost (XGB)

Extreme Gradient Boosting (XGB) is a tree-based supervised method in the area of machine learning. With Decision Trees as their base estimators, XGB methods build the former employing residuals [31]. Determination of how the nodes are split is carried out using the metrics Similarity Score (SS) and Gain (G) respectively depicted below in equations (3) and (4).

$$SS = (\sum_{i=1}^{n} Residual_i)^2 / (\sum_{i=1}^{n} [PP_i * (1 – PP_i)] + \lambda) \quad (3)$$

For the above equation (3),
– Residual is the real value that was seen or anticipated,

– the likelihood of an occurrence determined in a prior stage is known as the Previous Probability (PP), and

– Lambda is a Regularization parameter.

$$Gain = LL_{similarity} + RL_{similarity} – RT_{similarity} \quad (4)$$

For the above equation (4), while LL depicts the Left Leaf and the Right Leaf is depicted as RL, the root is denoted by RT respectively.

7.4 Gradient Boosting Classifier (GBC)

The Gradient Boosting Classifier (GBC) builds classification models in sequence. The subsequent models in turn try to reduce the errors of its preceding model. In other words, the predictors are built sequentially [32]. Consequently, the method is able to reach prediction values that closer to the original in comparatively lesser time because, the subsequent predictors gain knowledge from the mistakes of their predecessors [33]. GBC is a more extrapolated version of Extreme Gradient Boost talked about in the preceding sub-section.

7.5 Extremely Randomized Trees (ERT)

Extremely Randomized Trees (ERT) ensemble method where predictions are made based on averaging the predictions of an ensemble of trees which are built in a random manner [34]. Each one of these random trees are built by selecting at each node a number (for instance, K) of random splits and keeping the one that maximizes the score. This implies random selection of variable x_i and also that of threshold τ. These random trees are built till each one of the sub-samples present at all the end nodes or leaf nodes are either pure in terms of outputs of contains learning samples lower than the designated n_{min} [35].

Using larger values of K leads to trees preferentially using inputs with higher score values while using higher values of n_{min} results in smaller trees; the ensemble can be fined tuned using any one or both of these adjustments. Furthermore, tuning the optimal values for these two parameters are very much decided based on the problem to be solved.

8. Performance Measures

Notwithstanding the fact that accuracy is a common measure of performance for most of the authentication methods employed at large in the various research works, it may result in a single-sided performance appraisal perspective. Therefore, for providing the complementary perspectives, this work proposes to utilize a 4-tuple performance metric set proposed by Sokolova [36] for the methods used. The same are defined below.

– True Positives (TP) is ideally defined as the number of correctly recognized class examples corresponding to equation (5).

– True Negatives (TN) can be placed as the number of correctly recognized examples that do not belong to the class denoted by equation (6).

– False Positives (FP) is those examples that were incorrectly assigned to the class positive class illustrated by equation (7).

– False Negatives (FN) sums up the positive examples that were not recognized as positive class examples depicted by equation (8).

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

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

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

$$\text{F1 Score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) \quad (8)$$

9. Results and Discussions

Based on the nature of the original dataset wherein each one of the parameter values exists in different scales, a number of methods not exceeding five were chosen. This is owing to the

concept that they were less likely to be affected by this difference of scale and also their robustness in the direction of handling raw data.

The original un-normalized dataset D_{RAW} was fixated for training and testing all these five methods. For each user u_j a distinguished dataset D_j^S was deduced from D_{RAW} applying synthesizing factor Y. To remove any bias 10-fold cross validation [37], training and testing sessions were carried out using all the mentioned methods. The performance results produced from this are depicted in Table 1.

Table 1. Performance Results of the 5 methods.

	Decision Tree (DT)	**Random Forest (RF)**	**Gradient Boosted Classifier (GBC)**	**Extreme Gradient Boosting (XGB)**	**Extremely Randomized Trees (ERT)**
Accuracy	0.8489	0.8746	0.8838	0.8910	0.8910
Precision	0.8613	0.8935	0.8887	0.8971	0.8878
Recall	0.8396	0.8551	0.8845	0.8898	0.9061
F1 Score	0.8421	0.8652	0.8791	0.8868	0.8896

This work uses the swipe characteristics data or swipe-vectors in its raw form. This means the data in the various parameters are of different scales. In normal situations wherein machine learning techniques are used on this type of data with heterogeneous scales, the results of classification will be biased [38]. The data needs to be treated with techniques which may not be influenced by this variation of scale. Therefore, in the experimental setup a number of decision tree based methods are employed. These methods are refined and customized for the experiments.

All the five methods included in the experiments exhibited acceptable performance scores above 80 percent in all the metrics which include accuracy, precision, recall and F1 scores, respectively. A parameter wise comparison is done for the experiments. The maximum accuracy score is obtained by Extremely Randomized Trees with 89.10%. The highest precision score is bagged by Extreme Gradient Boosting at 89.71%. The scores of recall and F1 respectively are achieved by Extremely Randomized Trees as 90.61% and 88.96%.

The related works that were discussed in section 4 include two types of methods, namely those that used processed or scaled data and also those that did not. It is worth mentioning that, most of the methods did not mention the performance scores in terms of the 4-tuple performance metric set mentioned in section 9.

10. Conclusions

Touch based Continuous Authentication is a security method that is useful for perpetually validating the identity of a user. In the context of smartphones, this can be done unobtrusively by extracting hidden patterns from the swipe characteristics extracted from the touch gestures of a user on the touchscreen surface of the device. However, most methods require this raw data which include multiple parameters to be pre-processed and scaled to a homogenous form to be of any use for prediction. Further, most of the methods require a user to input more than one swipe to be able to correctly authenticate the same.

The work in this paper is twofold. Firstly, it deals with un-pre-processed or raw data. Secondly, it attempts to authenticate users based on single swipe for a given user. In comparison to coeval techniques, the Extremely Randomized Trees based method resulted in decent performance metrics comprising of 89.10% accuracy, 88.78% precision, 90.61% recall and 88.96% F1, individually.

References

[1] Lin, Z., Meng, W., Li, W. & Wong, D., (2020), "Developing Cloud-Based Intelligent Touch Behavioral Authentication on Mobile Phones," Deep Biometrics, pp. 141-159, Available: 10.1007/978-3-030-32583-1_7 [Accessed 20 March 2022].

[2] A. Orphanides & C. Nam, (2017), "Touchscreen interfaces in context: A systematic review of research into touchscreens across settings, populations, and implementations," Applied Ergonomics, vol. 61, pp. 116-143, Available: 10.1016/j.apergo.2017.01.013.

[3] Bhuyan, R., Kenny, S., Borah, S., Mishra, D. & Das, K., (2020), "Recent Advancements in Continuous Authentication Techniques for Mobile-Touchscreen-Based Devices," Smart Innovation, Systems and Technologies, pp. 263-273, Available: 10.1007/978-981-15-5971-6_29 [Accessed 20 March 2022].

[4] Frank, M., Biedert, R., Ma, E., Martinovic, I. & Song, D., (2013), "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 136-148, Available: 10.1109/tifs.2012.2225048.

[5] Stylios, I., Kokolakis, S., Thanou, O., & Chatzis, S. (2021), "Behavioral biometrics & continuous user authentication on mobile devices: A survey," Information Fusion, vol. 66, pp. 76-99, Available: 10.1016/j.inffus.2020.08.021.

[6] Jain, A., Bolle R., & Pankanti, S., (2006), Biometrics. New York: Springer.

[7] Mondal S., & Bours, P., (2017), "A study on continuous authentication using a combination of keystroke and mouse biometrics," Neurocomputing, vol. 230, pp. 1-22, Available: 10.1016/j.neucom.2016.11.031.

[8] Neal T., & Woodard, D., (2016), "Surveying Biometric Authentication for Mobile Device Security," Journal of Pattern Recognition Research, vol. 11, no. 1, pp. 74-110, Available: 10.13176/11.764.

[9] Ponce, A., (2015), "A Dynamic Behavioral Biometric Approach to Authenticate Users Employing Their Fingers to Interact with Touchscreen Devices," PhD Thesis, Graduate School of Computer and Information Sciences, Nova Southeastern University. Retrieved from NSUWorks, https://nsuworks.nova.edu/gscis_etd/46.

[10] Acien, A., Morales, A., Vera-Rodriguez, R., & Fierrez, J., (2020), "Smartphone Sensors for Modeling Human-Computer Interaction: General Outlook and Research Datasets for User Authentication," 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Available: 10.1109/compsac48688.2020.00-81 [Accessed 20 March 2022].

[11] Hochreiter, S., & Schmidhuber, J., (1997), "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, DOI: 10.1162/neco.1997.9.8.1735.

[12] Liu, J., Shahroudy, A., Xu, D., & Wang, G., (2016), "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," In: B. Leibe, J. Matas, N. Sebe, M. Welling (eds), Computer Vision – ECCV 2016, Lecture Notes in Computer Science, Vol. 9907, Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-46487-9_50.

[13] Debard, Q., Wolf, C., Canu, S., & Arne, J., (2018), "Learning to Recognize Touch Gestures: Recurrent vs. Convolutional Features and Dynamic Sampling," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 114-121, DOI: 10.1109/FG.2018.00026.

[14] Choi, S., Chang, I., & Teoh, A.B.J., (2018), "One-class Random Maxout Probabilistic Network for Mobile Touchstroke Authentication," 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3359-3364, DOI: 10.1109/ICPR.2018.8545451.

[15] Sitová, Z., (2016), "HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 5, pp. 877-892, May 2016, DOI: 10.1109/TIFS.2015.2506542.

[16] Samet, S., Ishraque, M., Ghadamyari, M., Kakadiya, K., Mistry, Y., & Nakkabi, Y., (2019), "TouchMetric: A Machine Learning Based Continuous Authentication Feature Testing Mobile Application," International Journal of Information Technology, 11(4), pp. 625-631, Available: 10.1007/s41870-019-00306-w.

[17] Zheng, A., & Casari, A., (2018), "Feature Engineering For Machine Learning", 1st ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472, United States of America: O'Reilly Media, Inc., pp. 29-33.

[18] Tornai, K., & Scheirer, W.J., (2019), "Gesture-based User Identity Verification as an Open Set Problem for Smartphones," 2019 International Conference on Biometrics (ICB), pp. 1-8, DOI: 10.1109/ICB45273.2019.8987373.

[19] Yang, Y., Guo, B., Wang, Z., Li, M., Yu, Z., & Zhou, X., (2019), "BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics", Ad Hoc Networks, vol. 84, pp. 9-18, Available: 10.1016/j.adhoc.2018.09.015 [Accessed 20 March 2022].

[20] Sharma, V., & Enbody, R., (2017), "User authentication and identification from user interface interactions on touch-enabled devices", Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Available: 10.1145/3098243.3098262 [Accessed 20 March 2022].

[21] Barlas, Y., Basar, O.E., Akan, Y., Isbilen, M., Alptekin., G.I., & Incel, O.D., (2020), "DAKOTA: Continuous Authentication with Behavioral Biometrics in a Mobile Banking Application," 2020 5th International Conference on Computer Science and Engineering (UBMK), pp. 1-6, DOI: 10.1109/UBMK50275.2020.9219365.

[22] Zhang, X., Zhang, P., & Hu, H., (2021), "Multimodal Continuous User Authentication on Mobile Devices via Interaction Patterns," Wireless Communications and Mobile Computing, vol. 2021, pp. 1-15, Available: 10.1155/2021/5677978.

[23] Syed, Z., Helmick, J., Banerjee, & S., Cukic, B., (2019), "Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability", Journal of Systems and Software, vol. 149, pp. 158-173, Available: 10.1016/j.jss.2018.11.017.

[24] Gunn, D.J., Roy, K., & Bryant, K., (2018), "Simulated Cloud Authentication Based on Touch Dynamics with SVM," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 639-644, DOI: 10.1109/SSCI.2018.8628762.

[25] Leingang, W., Gunn, D., Kim, J.H., Yuan, X., & Roy, K., (2018), "Active Authentication Using Touch Dynamics," South East Con 2018, pp. 1-5, DOI: 10.1109/SECON.2018.8479298.

[26] Shankar, V., Singh, K., (2019), "An Intelligent Scheme for Continuous Authentication of Smartphone Using Deep Auto Encoder and Softmax Regression Model Easy for User Brain," in IEEE Access, vol. 7, pp. 48645-48654, DOI: 10.1109/ACCESS.2019.2909536.

[27] Belman, A.K., Wang, L., Iyengar, S.S., Sniatala, P., Wright, R., Dora, R., Baldwin, J., Jin, Z., & Phoha, V.V., (2019), "Insights from BB-MAS - A Large Dataset for Typing, Gait and Swipes of the Same Person on Desktop, Tablet and Phone," arXiv:1912.02736.

[28] Shannon, C.E., (1948), "A mathematical theory of communication," in The Bell System Technical Journal, vol. 27, no. 4, pp. 623-656, Oct. 1948, doi: 10.1002/j.1538-7305.1948.tb00917.x

[29] Jijo, B.T., & Abdulazeez, A.M., (2021), "Classification Based on Decision Tree Algorithm for Machine Learning", JASTT, vol. 2, no. 01, pp. 20 - 28, Mar. 2021.

[30] Breiman, L., (2001), "Random Forests", Machine Learning, Vol. 45, pp. 5–32. https://doi.org/10.1023/A:1010933404324.

[31] Chen, T. , & Guestrin, C., (2016), "XGBoost: A Scalable Tree Boosting System". KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2016, pp. 785–794, DOI: https://doi.org/10.1145/2939672.2939785.

[32] Aziz, N., Akhir, E.A.P., Aziz, I.A., Jaafar, J., Hasan, M.H., & Abas, A.N.C., (2020), "A Study on Gradient Boosting Algorithms for Development of AI Monitoring and Prediction Systems", 2020 International Conference on Computational Intelligence (ICCI), 8-9 October 2020, Universiti Teknologi PETRONAS (UTP).

[33] Natekin, A., & Knollm, A., (2013), "Gradient Boosting Machines, A Tutorial", Frontiers in Neurorobotics, December 2013, Volume 7, Article 21, doi: 10.3389/fnbot.2013.00021.

[34] Geurts, P., Ernst, D., & Wehenkel, L., "Extremely Randomized Trees", Machine Learning, vol. 63, no. 1, pp. 3-42, (2006). Available: 10.1007/s10994-006-6226-1 [Accessed 20 March 2022].

[35] Wehenkel, L., Ernst, D., & Geurts, P., (2006), "Ensembles Of Extremely Randomized Trees And Some Generic Applications," In Proceedings of Robust Methods for Power System State Estimation and Load Forecasting.

[36] Sokolova, M., & Lapalme, G., (2009), "A Systematic Analysis Of Performance Measures For Classification Tasks", Information Processing & Management, Vol. 45, Issue 4, pp. 427-437, ISSN:0306-4573, DOI: https://doi.org/10.1016/j.ipm.2009.03.002.

[37] Refaeilzadeh, P., Tang, L., & Liu, H., (2009), "Cross-Validation," In: L. Liu, M. T. Özsu (eds), Encyclopedia of Database Systems, Springer, Boston, MA, DOI: https://doi.org/10.1007/978-0-387-39940-9_565.

[38] Bhuyan, R., (2020), "Contemporary Linear Stochastic Models for Forecasting IoT Time Series Data," Lecture Notes in Networks and Systems, pp. 99-106, Available: 10.1007/978-981-15-1624-5_10 [Accessed 20 March 2022].

Authors

Mr. Rupanka Bhuyan

Mr. Rupanka Bhuyan is a research scholar at St. Joseph University, Nagaland and currently the Registrar at ICFAI University, Nagaland. He has been involved in academics for more than two decades in the areas of Academic Administration, Software Development and Teaching computer science at postgraduate and undergraduate levels at various institutes, colleges and universities. His areas of interest in research are soft computing, machine learning and computer security.

Dr. S Pradeep Kumar Kenny

Dr. S Pradeep Kumar Kenny currently works as a Professor and Head of the Department of Computer Science under the Faculty of Engineering at St. Joseph University, Nagaland. He possesses M.Sc., M.Tech. and Ph.D. degrees in Computer Science and has over 15 years of teaching experience at undergraduate and postgraduate levels. His research interests span in the field of Image Processing and Artificial Intelligence.