International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No. 44, May 2024 – www.ijais.org

Machine Learning for User-Authentication on Mobile Devices using Typing Patterns

Odeniyi O.A.
Federal University of Technology, Akure
Department of Cybersecurity

ABSTRACT
With the rising cases of security issues, it has become a major concern and a matter worthy of note that having a reliable way of ensuring safe activities online or offline is an essential commodity in our present day. It is therefore important to consider safety a priority as individuals use their smartphones and mobile devices since these devices have come to be part of our lives. This research therefore focuses on how what an individual has, that is, how a behavioral pattern which is unique to every individual can be harnessed to ensure security and privacy of data and information. In doing this, Decision Tree (DT) and K-Nearest Neighbor (KNN), both machine learning algorithms were implemented and used to analyze the features of different people's typing patters. A static password was used and every subject was required to type the password into a smartphone in order to capture their typing features. Afterwards, the required features were extracted and further analyzed for the purpose of use for security. At the end of our experiments, the results came out with an accuracy of 99.12% and 99.92% from KNN and DT respectively.

General Terms
Pattern Recognition, Security, Algorithms, Machine Learning,

Keywords
Decision Trees, K-Nearest Neighbor, Recall, Accuracy, False Positive

1. INTRODUCTION
Typing patterns, also known as keystroke dynamics has gained attention as a means of enhancing smartphone security. Even though most people have come to realize there is need for personal responsibility in ensuring the safety of one's devices, not many people have embraced the required steps needed to achieve it. Even though some people use passwords to secure their devices, majority are not mindful of the effect that the strength of such passwords have on the security of their devices.

In view of the reality of security issues, typing patterns, also known as keystroke dynamics are becoming more popular by the day as a means of further securing devices and equipment. [1]. This is possible by analyzing the way an individual types in order to retrieve features like typing speed, rhythm, and pressure; thereby creating unique user profiles for an extra layer of security.

Smartphones have become a major part of human existence and are widely used for a large variety of activities: from social networking to online shopping, from message exchanging to mobile gaming, to mention just a few [2]. Mobile phones have become part of our daily lives and besides communication, most people engage in different activities on their devices which generate private information that are sometimes required to be stored on the phone. In fact, data is the most expensive commodity today, hence the need to protect it cannot be over-emphasized. [2]

2. AIM AND OBJECTIVES
The specific objectives of the research work are to:

(a) design a keystroke user-authentication model for mobile devices using k-nearest neighbor and decision tree;

(b) implement the model designed in (a); and

(c) evaluate the model designed in (a).

3. METHODOLOGY
Keystroke dynamics, keystroke biometrics, typing dynamics and lately typing biometrics, is the detailed timing information, which describes exactly when each key was pressed, and when it was released as a person is typing at a keyboard.

The model of the entire system is a combination of series of steps, phases, techniques and algorithms. At the different stages, several varying techniques are tested and used in order to get effective and efficient answers to the questions that this research seeks to tackle and address. The conceptual models of the system comprise the enrolment model where each user's template is created while the login module handles the decision-making process of confirming if an individual is given access to the device. A score is announced when the user's login template matches the claimed stored template.

3.1 Data Collection

This first stage of the system architecture is very essential and sensitive because it will determine the overall performance of the system. Hence, there is need to ensure high quality data capture at this level. This is where individual typing data are collected for training, testing and validation of the system. In essence, an Android-based data-collection mobile application was developed. The Android platform was used because of its widespread use and affordability, compared to other mobile platforms.
After deploying the application on the Android device, different people were approached in order to capture their typing dynamics.

As stated earlier, a static password was used- which means that every subject whose typing pattern was collected typed the same password. A data collection application was developed and installed on the device. The application is comprised of four interfaces where different activities were carried out for the data collection process: Information page, Settings page, Typing Dynamics page and Timings page. The total number of variables or features retrieved at each instance of typing the chosen password phrase is expressed with the following

notation:

$$N = (2(H + T) - 1) \quad (1)$$

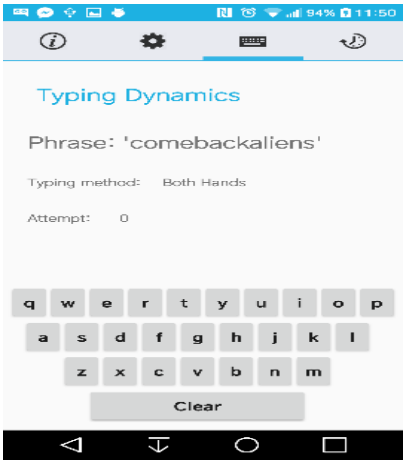

Fig 1: The Typing Dynamics page

3.2 Data Pre-Processing

Data preprocessing involves preparing data in a format that will be suitable to the classification algorithm. This process improves classification performance and also reduces the computational time in training and testing the Machine learning algorithms.

The following steps were carried out in the course of processing the keystroke data.

- Dropping of irrelevant features: Some features that obviously do not contribute significantly to keystroke dynamics' classification were expunged from the data.
- Non-numeric to Numeric feature conversion: Keystroke data contain mixed feature types (categorical and numeric features) and the classification algorithms used in this research can only work or perform better on numeric features. Hence, the categorical features were converted to numeric.
- Feature Normalization /Scaling: feature normalization or scaling comes in situations where there is high variation among feature values. This process is necessary to avoid bias problem or issues during classification. In this experiment, min-max normalization was employed to scale high varied feature values into comparable range of zero (0) and one using the equation below

$$v' = \frac{v - min_f}{max_f - min_f} \quad (2)$$

where v' represents the new value, v denotes the observed value (that is, the value to be normalized), max_f and min_f are maximum and minimum values of feature f respectively. The procedure includes stripping the class label (the last column) off the datasets before carrying out the method.

3.3 Data Classification

This next stage involves training of the dataset with 2 different machine learning algorithms – Decision Trees (DT) and K-Nearest Neighbor (KNN). Data classification is the process of organizing data into categories that make it easy to retrieve, sort and store for future use [3]. This is where training and testing of the proposed keystroke dynamics predictive system were carried out. Data classification is the process of separating and organizing data into relevant groups ("classes") based on their shared characteristics, such as their level of sensitivity, the risks they present, and the compliance regulations that protect them. To protect sensitive data, it must be located, classified according to its level of sensitivity, and accurately tagged [3]. In this research, the test dataset (without class labels) were fed to the machine learning algorithm and prediction for each row in the test dataset was obtained Thereafter, a comparison between the expected outcomes and the predicted outcomes were performed by counting the number of correct and incorrect predictions for each class. These numbers are then organized into a table where each row corresponds to the actual class and each column corresponds to a predicted class. The counts of correct and incorrect classification are then filled into the table where all correct classifications lie along the principal diagonal and the incorrect classifications correspond to numbers off the diagonal.

The standard metrics used in validating the predictive model include Precision, Recall, and f1-score. These metrics were computed from True Positives (TP), False Negatives (FN), False Positives (FP) and True Negatives (TN) obtained through confusion matrix. The TP, FN, FP, and TN were extracted from the confusion matrix table as follows:

		Predicted class	
		Class = Yes	Class = No
Actual Class	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Fig 2: Relationship of True Positive, False Negative, False Positive and True Negative

3.3.1 Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observations to the total observations. Accuracy is a great measure but only when you have symmetric datasets where values of false positives and false negatives are almost the same. Therefore, you have to look at other parameters to evaluate the performance of your model.

Accuracy = TP+TN/TP+FP+FN+TN (3)

3.3.2 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answers is "of all individuals that were identified, how many were actually and truly identified for authentication?" High precision relates to a low false positive rate.

Precision = TP/TP+FP (4)

3.3.3 Recall

Recall is the ratio of correctly predicted positive observations to all observations in actual class - yes. The question recall answers is: "Of all the passengers that were truly identified, how many were labelled?"

Recall = TP/TP+FN (5)

3.3.4 F1 Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand

as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision) (6)

By applying these parameters from the confusion matrix, the following performance measurements were derived:

$$Accuracy = \frac{TP+TN}{Total\ number\ of\ all\ test\ entries} * 100 \quad (7)$$

$$Precision_c = \frac{TP_c}{TP_c + FP_c} * 100 \quad (8)$$

$$F1 - measure_c = 2 * \frac{Precision_c * Recall_c}{Precision_c + Recall_c} * 100 \quad (9)$$

where c represents the class label, TP_c represents number of correctly predicted class label, while FP_c and FN_c are the number of a particular class that is misclassified. However to obtain the overall performance of the system, the average of all precision, recall, and f1-measure were obtained using micro and macro average methods respectively.
A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

3.3.5 Micro Average method

Micro Average method is used in this research to view the effect of imbalance in the dataset. It is computed by adding up all TP, FP, and FN for each class label applied them to obtain precision and recall as shown below:

$$Precision = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c} * 100 \quad (10)$$

$$Recall = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c} * 100 \quad (11)$$

3.3.6 Macro average method

Macro average independently takes the precision and recall of each class label and finds their average. This was applied to determine the performance of the model across the classes. It can be used when you want to know how the system performs overall across the sets of data. Even though it is not relevant for making any specific decision it can be a useful measure when the dataset varies in size. Macro-averaged metrics are used to evaluate systems performance across on different datasets.

Macro-Precision = $\frac{Precision1+Precision2}{2}$ (13)

Macro-Recall = $\frac{Recall1+Recall2}{2}$ (14)

Macro-F1-Score =2. $\frac{RMacro-Precision.Macro-Recall}{Macro-Precision+Macro-Recall}$ (15)

3.4 Decision Tree (DT)

Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems. [4] Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. [5] The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Afterwards, DT makes prediction $V_2(X)$ by iteratively partitioning the hand stroke training set T into j subsets $(t_1, t_2, \dots, t_j)$ where j is the number of outcome of test over particular feature f_i. The process is continued over each t_k, where $1 \leq k \leq j$, until all elements in each final subset fall under the same class (identity of keystrokes). Information gain presented in (1) will be employed to determine the best feature to divide the subset at each stage.

$$IG(T,f) = I(T) - \sum_{v \in values(f)} \frac{|T_v|}{N} I(T_v) \quad (16)$$

where v is a value in feature f, $values(f)$ represents all possible values in f, T_v represents instances for which f has v, N represents number of instances in T, $I(T)$ and $I(T_v)$ represents entropy of T *and* T_v respectively.

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. It is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. They are adaptable at solving any kind of problem at hand.

In Decision Tree Learning, a new example is classified by submitting it to a series of tests that determine the class label of the example. These tests are organized in a hierarchical structure called a decision tree.

Entropy=- $\sum P_k \, log P_k$ (17)

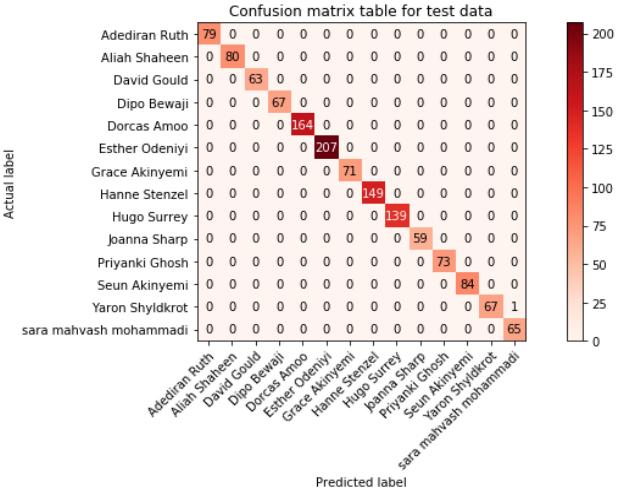

Fig 3: Confusion matrix table on test data for Decision Tree (DT)

3.5 K-Nearest Neighbours (KNN)

KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions) [6]. KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. KNN is a non-parametric supervised learning technique in which data point is classified to a given category with the help of a training set. In simple words, it captures information of all training cases and classifies new cases based on a similarity. The KNN algorithm is one of the simplest classification algorithms. Even with such simplicity, it can give highly competitive results [7]. KNN algorithm can also be used for regression problems, but it was used for classification in this research work.

Given a new or unlabeled keystroke instance, the algorithm will find k (number of neighbours) points in the keystroke training set that are closest to the unlabeled keystroke instance. This is determined using Euclidean distance d between instances in the training set and the unlabeled keystroke instance with features f_i and q_i respectively as follows:

$$d = \sqrt{\sum_{i=1}^{p}(f_i - q_i)^2} \qquad (18)$$

where p represents the total number of hand stroke data features. Then, the majority label vote will be selected among classification of the k points as the prediction result $V_1(X)$ of new key stroke instance. If k=1, then the case is simply assigned to the class of its nearest neighbor, though most times using k=1 returns the case in view. One of the advantages of nearest-neighbor classification is its simplicity. To maximize the advantage of this algorithm, different distance functions can be used but they are dependent on the type of variables in use. For example, Euclidean, Manhattan and Minowski distances are only valid for continuous variables while Hamming distance is used in the instance of categorical variables. There is also a need for standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

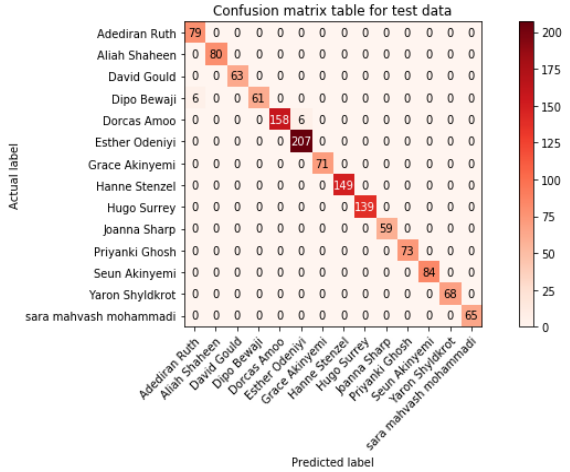

Fig 4: Confusion matrix table on test data for K-Nearest Neighbours (KNN)

Table 1 Comparison of DT model and KNN model using macro average metrics

Models	Precision	Recall	F1	Accuracy
DT	99.86	99.93	99.86	99.92
KNN	99.29	99.07	99.14	99.12

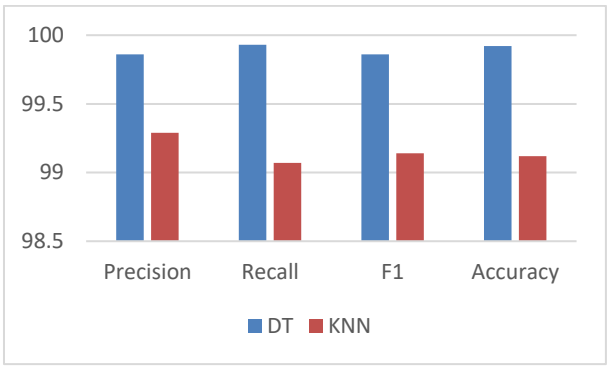

Fig 5: Comparison of DT model and KNN model using macro average metrics

The keystrokes data collected from individuals were taken through some preprocessing stages to make them fit for use in the developed system.

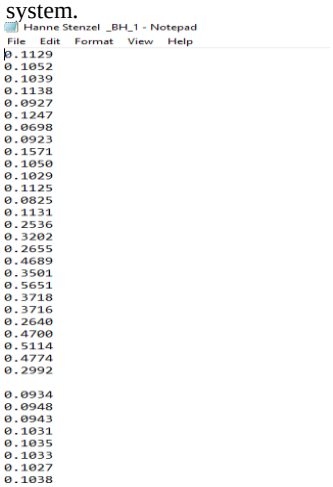

Figure 6 Sample Raw Hold time and Flight time data obtained from an individual

4. CONCLUSION

In conclusion, this research shows that using different algorithms helps to know and compare which one is better than the other. From the experiments that were run, the results from Decision Trees (DT) showed better performance overall, compared to K-Nearest Neighbours (KNN). This therefore means that DT can be used alone because in all the evaluated metrics, it out-performed KNN. However, further research can be done in the by combining the strengths of different algorithms in order to have a better-performing system. Also, other algorithms can be experimented in order to further improve user authentication.

5. REFERENCES

[1] Simon Parkinson, Saad Khan, Alexandru-Mihai Badea, Andrew Crampton, Na Liu and Qing Xu (2022), An empirical analysis of keystroke dynamics in passwords: A longitudinal study, https://doi.org/10.1049/bme2.12087

[2] Tiago Dias, João Vitorino, Eva Maia, Orlando Sousa, Isabel Praça (2023) KeyRecs: A keystroke dynamics and typing pattern recognition dataset, Data in Brief, Volume 50, 2023, 109509, ISSN 2352-3409

[3] Altwaijry, N. (2023) Authentication by Keystroke Dynamics: The Influence of Typing Language. Appl. Sci. 2023, 13, 11478. https://doi.org/10.3390/app132011478

[4] Charbuty Bahzad and Mohsin Abdulazeez Adnan. (2021), Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends, 2. 20-28. 10.38094/jastt20165.

[5] Kohavi Ronny and Quinlan Ross (2002), Data mining tasks and methods: Classification: Decision-tree discovery. Handbook of Data Mining and Knowledge Discovery, 267-276.

[6] Oluwatobi Ayodeji Akanbi, Iraj Sadegh Amiri and Elahe Fazeldehkordi (2015), A Machine-Learning Approach to Phishing Detection and Defense, Syngress, 2015, Pages 35-43, ISBN 9780128029275, https://doi.org/10.1016/B978-0-12-802927-5.00003-4.

[7] Alsammak, Ihab & Sahib, Humam, H.Itwee and Wasan. (2020), An Enhanced Performance of K-Nearest Neighbor (K-NN) Classifier to Meet New Big Data Necessities, IOP Conference Series: Materials Science and Engineering, 928. 032013. 10.1088/1757-899X/928/3/032013.