




Combining behavioral biometrics and session context analytics to enhance risk-based static authentication in web applications

Jesus Solano^{1,2} · Luis Camacho^{1,2} · Alejandro Correa² · Claudio Deiro² · Javier Vargas² · Martín Ochoa^{1,2} 

© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The fragility of password-based authentication has been recognized and studied for several decades. It is an increasingly common industry practice to profile users based on their sessions context, such as IP ranges and Browser type in order to build a risk profile on an incoming authentication attempt. On the other hand, behavioral dynamics such as mouse and keyword features have been proposed in the scientific literature order to improve authentication, but have been shown most effective in continuous authentication scenarios. In this paper we propose to combine both fingerprinting and behavioral dynamics (for mouse and keyboard) in order to increase security of login mechanisms. We do this by using machine learning techniques that aim at high accuracy, and only occasionally raise alarms for manual inspection. We evaluate our approach on a dataset containing mouse, keyboard and session context information of 24 users and simulated attacks. We show that while context analysis and behavioural analysis on their own achieve around 0.7 accuracy on this dataset, a combined approach reaches up to 0.9 accuracy using a linear combination of the outcomes of the single models.

Keywords Behavioral dynamics · Static authentication · Machine learning

1 Introduction

Several studies have pointed out the challenges that password-based authentication pose for robust security [1, 2]. With the increasing popularity of web services and cloud-based applications, we have also seen an increase on attacks to those platforms in the past decade. Several of those

publicly known attacks have involved stealing of authentication credentials to services (see for instance [3]). In addition to this, passwords are often the target of Malware (for instance banking related Malware such as Zeus [4] and its variants). So even if one would assume users are forced to select strong passwords (from the point of view of difficulty to guess), password-based authentication does not provide strong security guarantees.

In order to mitigate the risk posed by attackers impersonating legitimate users by means of compromised or guessed credentials, many applications use mechanisms to detect anomalies by analyzing the connection features such as incoming IP, browser and OS type as read by HTTP headers, among others. Some of these context-based features have been also been discussed in the scientific literature [5]. However, there are some limitations of those defensive mechanisms, for example, if the anomaly detection is too strict, there could be false positives that would harm user experience and thus hurt the webservices from a business perspective. On the other hand, if a careful attacker manages to bypass such context-related filters, for instance by manipulating HTTP parameters, using VPN services, or ultimately using a victim's machine [6], then such countermeasures fall short to provide better security.

✉ Martín Ochoa
martin.ochoa@appgate.com

Jesus Solano
jesus.solano@appgate.com

Luis Camacho
luis.camacho@appgate.com

Alejandro Correa
alejandro.correa@cyxtera.com

Claudio Deiro
claudio.deiro@cyxtera.com

Javier Vargas
javier.vargas@cyxtera.com

¹ AppGate Inc., Cra 13A # 98-75, Bogotá, Colombia

² Cyxtera Technologies, BAC Colonnade Office Towers, 2333 Ponce De Leon Blvd, Suite 900, Coral Gables, FL 33134, USA

Behavioral biometrics [7] have been proposed in the literature as a strategy to enhance the security of both web and desktop applications. They have shown to work with reasonable accuracy in the context of continuous authentication [8, 9], when both the training and the monitoring time of mouse and/or keyboard activity is long enough. In the context of static authentication, where interaction during log-in time with users is limited, such methods are less accurate and may be impractical [10], unless long static authentication interactions are assumed and many sessions are available for training. However, in today's internet of services, many websites rely on third parties for security related functionality, that is integrated in the form of external javascript snippets. In domains handling highly sensitive data such as banking, those services are often only allowed to interact with a user's session during or before log-in, but not post-login. Users log-in only sporadically and thus not sufficient training data is available to use some models that have been proposed in the literature. Therefore improving static risk-based authentication is a practical challenge.

Our proposed solution to address the above mentioned shortcomings of the individual context-based risk assessment techniques is to synergistically consider machine-learning based methods to detect anomalies in both context (browser type, country of origin of IP etc.) and behavioral features of a given user at login time. By considering a model that takes into account several features of browser, operating system, internet connection, connection times, keystroke and mouse dynamics one gains more confidence on the legitimacy of a given log-in attempt. Our model analyzes several previous log-in attempts in order to evaluate the risk of a new log-in attempt and is based on realistic data from customers of several major banks.

On the other hand, we build a lean machine-learning model that relies on data from the last 10 login attempts to give a score on the biometric behaviour, and thus can be used without large amounts of training data per user. These ideas have been preliminary explored in [11], which we extend in the following ways.

- We improved the methodology used to generate log-in attempts out of the TWOS [12] dataset and re-designed the evaluation of the proposed behavioural model in order to obtain more realistic data and to better assess how the approach generalizes.
- We discuss in depth the accuracy of the individual models in isolation, and their accuracy in a scenario that has both context and behavioural attacks.
- We explore three different metamodels of the individual models in order to find the best performing combination of parameters. The best combination achieves an accuracy of up to 0.9 in our dataset.

The rest of the paper is organized as follows: in Sect. 2 we recap some notions of context analytics and behavioral dynamics. In Sect. 3 we present our approach, and describe the data collected and the experimental design. In Sect. 4 we describe the experiments carried out in order to assess the effectiveness of the proposed approach. We discuss related work in Sect. 5 and conclude in Sect. 6.

2 Background and attacker model

User authentication has been traditionally based on passwords or passphrases which are meant to be secret. However, secrets can be stolen or guessed and, without further authentication mechanisms, attackers could impersonate a victim and steal sensitive information. To avoid this, the implementation of risk based authentication has allowed traditional authentication systems to increase confidence on a given user's identity by analyzing not only a pre-shared secret, but other features, such as device characteristics or user interaction which are expected to be unique [13, 14]. In the following we review some fundamental concepts related to device fingerprinting for authentication and behavioral biometrics.

2.1 Device fingerprinting

Device fingerprinting is an identification technique used both for user tracking and authentication purposes. The main goal of this technique is to gather characteristics that uniquely identify a device. There are different ways to create this profile, the most reliable of them involves creating an identifier based on hardware signatures. However, acquiring these signatures requires high level privileges on the device, which is often hard to achieve.

Thanks to the popularization of the internet and the increased browser capabilities it is possible to also use statistical identification techniques using information gathered from the web browser [5, 15], such as browser history, installed plugins, supported mime types, user agents and also network information like headers, timestamps, origin IP and geolocation. Geolocation can be either collected using HTML5 or approximated from an IP address by using appropriate services. Gathering only browser information means these techniques identify web browsers and not necessarily devices or users. On the other hand, parameters such as HTTP request parameters are easy to spoof. Most recent techniques try to combine both hardware and statistical analysis gathering the information using the web browser capabilities, these techniques use HTML5 [16] and javascript APIs to measure the execution time of common javascript functions and the final result of rendering images as hardware signatures, these measurements are

compared to a base line of time execution and rendering performed in a known hardware used as control [17, 18].

2.2 User behavior identification

Another popular risk-based authentication technique is behavioral analysis, based on mouse and keyboard dynamic statistics. The underlying idea of measuring user behavior is to turn human-computer interactions into numerical, categorical and temporal information. The standard interactions gathered for a behavioral model are key-strokes, mouse movements and mouse clicks. For instance, common features extracted from keyboard events are key pressed and key released events, together with their time-stamps. For mouse, cursor position, click coordinates and timestamps are commonly used [12]. Such features are processed and aggregated to profile user behavior. In this work, we will use aggregations such as the ones discussed in [19]. As shown in Fig. 1 we used the suggested space segmentation in [19] to calculate mouse movement features.

These behavioral features give us information about very unique characteristics of each user such as how fast the user types, how many special keys the user uses, what is the proportion of use of mouse and keyboard, how long the user stops interacting before finishing an activity. The intuition behind this is that it must be easy to distinguish a user who uses mainly mouse from a user who uses mainly keyboard, also intuitively some physical conditions like hardware and user's ability with the peripheral devices makes these interactions more unique.

Behavioral models use machine learning to identify users by using these feature vectors. Notice that by recording one user's interaction in the same situation many times, it is expected that this user will interact with the computer similarly each time and also that it differs from the interactions gathered from other users.

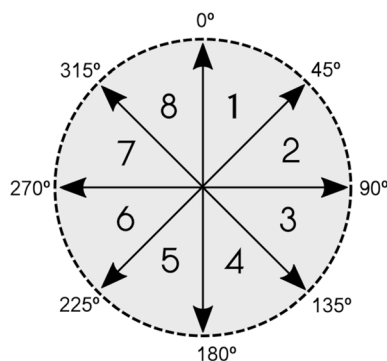


Fig. 1 Mouse directions segmentation

2.3 TWOS dataset

For the behavioral dynamics analysis, that we will illustrate in the following sections, it is important to have mouse and keyboard dynamics data, in order to evaluate our models. For this purpose, we have chosen to use data from a public data set known as *The Wolf Of SUTD (TWOS)* [12]. The data set contains realistic instances of insider threats based on a gamified competition. We have chosen this dataset since it contains both mouse and keyboard traces, among others. In [12], authors attempted to simulated user interactions in competing companies, inducing two types of behaviors (normal and malicious). The data set contains both mouse and keyboard data of 24 different users. We chose the TWOS dataset because of the large amount of behavioral patterns they recorded. In total, TWOS data set has more than 320 hours of mouse and keyboard dynamics. Data was continuously collected for volunteers during routine internet browsing activities in the context of a gamified experiment. The mouse agent collected the position of the cursor in the screen, the action's timestamp, screen resolution, the mouse action, and user ID. The mouse actions involved in our analysis are mouse movement, button press/release and scroll. The keyboard agent logged all characters pressed by the users. The data set includes the timestamp of event, movement type (press/release), key and user ID. Both alphanumeric and special keys were recorded by the agent. Since the users typed potentially sensitive information the data is provided in an anonymized fashion. The keyboard was divided into zones to accomplish the anonymization. Figure 2 shows the mapping of the keyboard into three zones to enhance the privacy concerns.

2.4 Attacker model

We assume an attacker that has gained access to a victim's credentials to authenticate to a webservice (login and password). An attacker may also gain knowledge about, or try to guess, the context in which a victim uses a service: the time of the day in which a user usually connects, the operating system used, the browser used and IP range from which a victim connects. We assume that an attacker could employ one of the following strategies, or more than one in combination with others to attempt to impersonate a victim:

$$\begin{aligned} \{I, O, P, J, K, L, N, M\} &\rightarrow \text{RIGHT} \\ \{R, T, Y, U, F, G, H, V, B\} &\rightarrow \text{CENTER} \\ \{Q, W, E, A, S, D, Z, X, C\} &\rightarrow \text{LEFT} \\ \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} &\rightarrow \text{DIGIT} \end{aligned}$$

Fig. 2 Keyboard mapping layout to anonymize sensitive information

- *Simple attack*: The attacker connects to the webservice from a machine different than the victim's machine.
- *Context simulation attack*: The attacker connects to the webservice from a machine different than the victim's machine, but tries to replicate or guess the victim's access patterns: OS, Browser type, IP range and time of the day similar to victim's access patterns.
- *Physical access to victim's machine*: An attacker connects from the victim's machine, thereby having very faithfully replicated a victim's context, and attempts at impersonating the victim.

Note that we explicitly exclude from the attacker's capabilities that of recording and attempting to replicate a victim's behavioral dynamics (keyboard and mouse usage features). We believe that although this is an interesting attacker model, it is an extremely powerful one, and we leave its treatment to future work.

3 Approach

The goal of our approach is to overcome the shortcomings of the single risk assessment strategies (context-based analysis of HTTP connections and behavioral dynamics) by obtaining a single model that takes into account both strategies.

In Table 1 we summarize the effectiveness of various strategies in detecting the attacks discussed in the previous section, and also highlight the desired outcome of our approach. In essence, we expect a combined model to perform better in case of attacks, given that the combined model can recognize both changes in context and changes in behavior. Note that in this table we assume there is always impersonation (and thus always changes in behavior).

Moreover, we highlight the potential misclassification of the various approaches in various scenarios in Table 2. Here, we summarize the expectation of the combination of both approaches in terms of reducing false positives. When a user uses a new device, one would expect its behavior to be similar in terms of keystrokes and mouse dynamics (although not exact). When he travels, it should remain very similar, those correcting possible false positives from the context analysis.

In the following we will summarize the models we used for the single risk-based strategies, and describe how these models are used in combination to produce a combined

risk-based assessment strategy. It is important to note that for the context analytics data we will assume that some users have a heterogeneous access pattern (i.e. from multiple devices and locations, due to travel), as depicted in Fig. 3 for a user for which we have 338 access records. On the other hand, the time of activity considered for behavioral interaction reflects the average time of a password based login (which typically is a value between 25 and 30 seconds). Because of these challenges single models are not perfect within a global context attack, but can be used in synergy to produce a better model as we will show in the evaluation section.

3.1 History-aware context analytics

In this subsection we describe the high-level construction of a session context model, based solely on session data obtained from HTTP requests. We assume users with complex access behaviors such as the ones depicted in Fig. 3, so we need to build a system that is good at detecting anomalies and potential attacks, but also it is somewhat flexible to certain changes in context that could be benign.

We assume a system that records usage statistics of the number of times that a user logs in, the day and time of the week at which the user logs in, what type of device and browser they are using, and the country and region from which the user is accessing. Currently, platform and browser data is obtained parsing the user agent, and geographic data is obtained parsing the IP address, information that can be obtained from network sessions corresponding to successful log-ins for a given user.

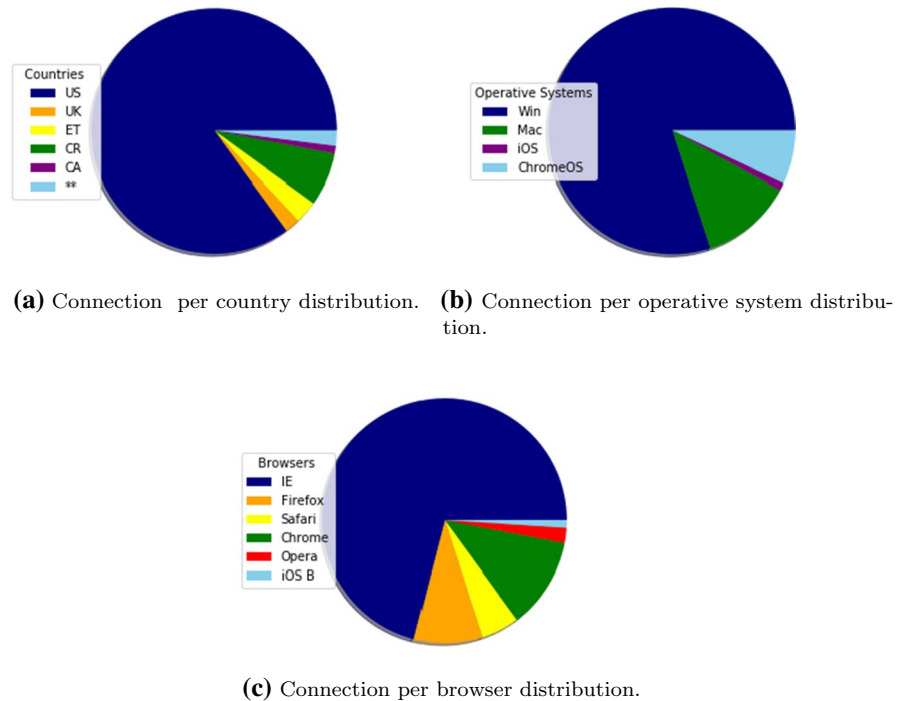
One of the challenges of building such a model is the fact that several categories considered are non-numerical (for instance a given browser version or operating system). This forces us to use a feature vector with connections statistics on each browser model version, each day of the week, each country and region etc. On the other hand, we

Table 2 Strategies versus benign context changes

Approach	New machine	User travels
Context analytics	Likely FP	Likely FP
behavioral dynamics	Likely accurate	Accurate
Combination	Likely accurate	Accurate

Table 1 Strategies versus attack vectors

Approach	Simple attack	Context simulation	Physical attack
Context analytics	Effective	Partially effective	Ineffective
behavioral dynamics	Partially effective	Partially effective	Partially effective
Combination	Effective	More effective than single approaches.	Partially Effective

Fig. 3 Context of user with heterogeneous access patterns

must somehow assess the likelihood of a given connection context in order to decide whether a new connection is anomalous or not. One way to do this is to simply compute the ratio of observations in a given field of a category divided by the sum of all the observations in that category.

For instance, let c the number of connections coming from a country K . Let C the total number of observations coming from all countries for a given users. Then the likelihood of an incoming connection from K could be computed as $\frac{c}{C}$. In order to assign a probability of 1 to the most likely event within a category, and a relative weight to other events in decreasing order from most likely to less likely, we normalize all values within a category as follows: order fields from most likely to less likely, define a new probability for a given field within a category as the sum of the probabilities for categories with probability equal or less to the one of the given field. For example, consider three countries with the following probabilities based on access frequency: $US = \frac{1}{2}$, $UK = \frac{1}{3}$, $FR = \frac{1}{6}$. The normalized probabilities would be: $US = 1$, $UK = \frac{3}{6}$ and $FR = \frac{1}{6}$.

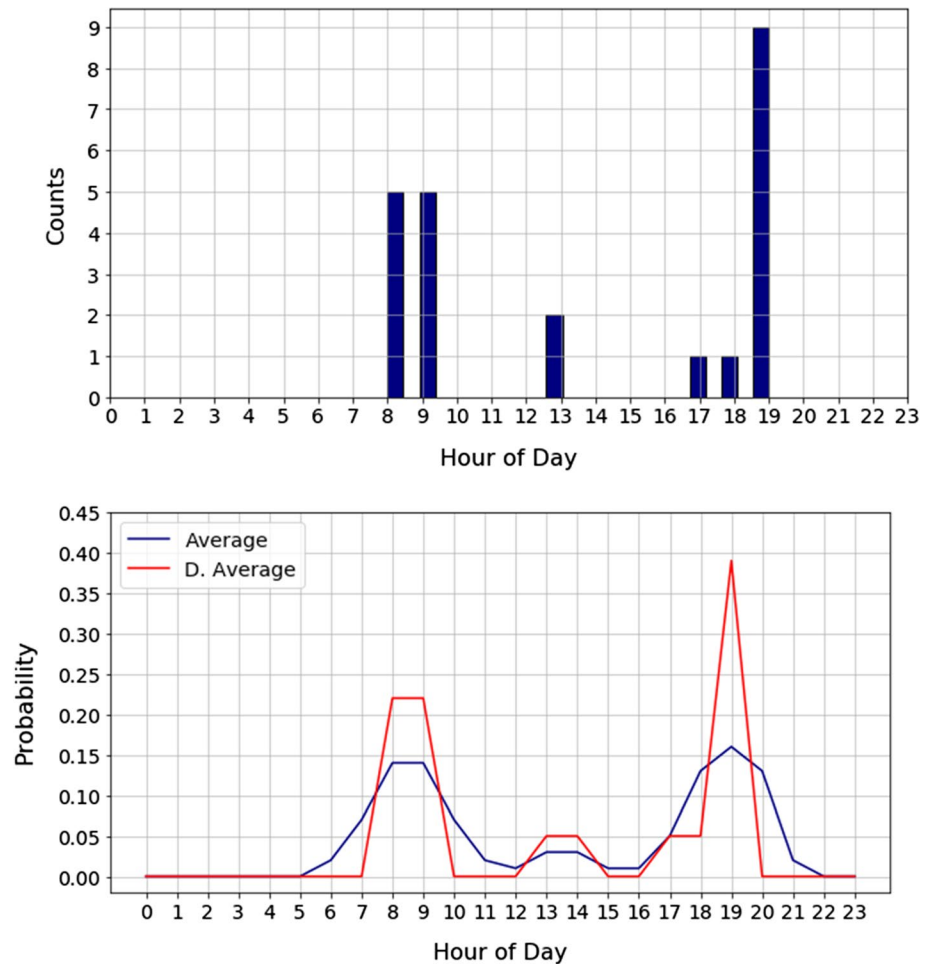
Moreover, temporal categories (hours, days, etc.) are considered cyclical, because for instance events around midnight (before or after 24:00) should be considered relatively close to each other. Also, in order to smooth the notion of 'closeness' in discretized events such as frequencies of access in different hours of the day, we use a convolution as depicted in Fig. 4. In this example, we have a distribution of discrete frequencies around the clock for a given user. In this scenario, 7PM is the hour of the day with most access. However this is close to say, 8PM, so it would be appropriate to consider an access at 8PM relatively normal for this context.

The feature vector for a session login attempt is formed using the normalized probability for each variable gathered from the HTTP request. For example, in the countries case above, a session which comes from US will have a value of 1 for variable country in the feature vector. To train the model we calculate the probability profiles for each user using the login history. Afterwards we evaluate a subset of new logins with the user probability profile and compute the feature vector for each visit. The feature vector is fed to a Random Forest model that assesses how anomalous the current event is. The impersonation records were synthesized comparing login events from one user to the history of another user. With this in mind, the model assesses the likelihood of an impersonation. Finally the statistics are updated, the idea being that the system will gradually adapt to permanent changes in the user profile.

3.2 Behavioral dynamics combining keystrokes and mouse activity

Both keyboard and mouse events are enough to describe a human-computer interactions and turn it into behavioral features. It is obvious that a regular user uses both at the same time. However, there is no simple way to merge both keyboard and mouse dynamics features. To describe a user behavior during a session we calculate the keyboard and mouse dynamics using all the gathered events in one single session, where a session is defined as a time frame where the user is performing any activity on the computer. Once the keyboard and mouse dynamics are calculated, we combine both set of features, resulting in only one single vector of

Fig. 4 Graphical representation of convolution used for temporal categories (e.g. hour of connection)



features per session. The combination of both set of features describes the use of keyboard and mouse dynamics in a single session. This process is repeated each time a new session is gathered. To compare a session behavior vector against the sessions in history we defined a maximum number of sessions to compare, in our experiment for each user we randomly chose between 10 and 30 sessions, this allows to test the algorithm performance with different history length. We calculated the history mean by using Eq. 1, as follows:

$$FeatureHistMean_j = \frac{\sum_j FeatureHist_j}{|J|} \quad (1)$$

Where $FeatureHistMean_j$ is defined as the the mean of one feature, $FeatureHist_j$ is the individual observation of the feature and J is the number of observations in the history. To compare the gathered session against the user sessions history we used Eq. 2.

$$FeatureDist_i = \frac{Feature_i - FeatureHistMean_i}{1 + \sigma(FeatureHist_i)} \quad (2)$$

Where $FeatureHistMean_i$ is the calculated mean of the feature and $\sigma(FeatureHist_i)$ is the feature standard deviation. The resulting vectors of deviations give us the distance of a session compared to the history.

Using the previously described behavioral analysis process, we created a data set of sessions with labeled data. To create the positive labels we calculated for each user a base history. Then we calculated the behavioral features and deviation vectors. To create the negative labels for each user we randomly selected different users sessions and ran the behavioral analysis against the original user history. The resulting vectors feed a random forest algorithm to assess if a session is legitimate or not.

3.3 Overview of combined model

Assume we have a model to assess the risk of a session based on the browser context information, and another model to identify users by using behavioral patterns. As discussed in the introduction, there are however inherent limitations to each of the single models: context-based info of an incoming network

(HTTP) connection cannot detect advanced impersonation attacks, whereas behavioral info is not accurate enough in short interactions such as log-ins. As a result we propose to enhance the risk-based authentication system's overall performance by combining the predictions of both models.

In principle, there are several ways to build such a combined meta-model, for instance by building a decision flow-chart that takes the scores produced by the singles models and decides whether a given session should be considered suspicious or not. In this work, we propose to study three different combination methods of both scores: (1) a parametric linear combination, (2) a random forest classifier and (3) a Support-vector Machine (SVM) classifier to predict the combined label of both scores. Let us to define $\hat{y}_c, \hat{y}_b \in [0, 1]$ as the prediction of context-based and behavioral model, respectively. First at all, we propose to unify the models' prediction using a linear convex combination as we describe in Eq. 3.

$$\hat{y}_t = \alpha_c \cdot \hat{y}_c + \alpha_b \cdot \hat{y}_b \quad (3)$$

where $\alpha_c, \alpha_b \in [0, 1]$ are the coefficient parameters of each model. Note the coefficients must satisfy $\alpha_c + \alpha_b = 1$, because to be a meaningful prediction $\hat{y}_t \in [0, 1]$. In the evaluation section we will discuss an example instance of the parameters. Second, we use a Random Forest classifier to predict the combined label. The random forest is fed with the prediction score of context-based and behavioral model. Third, we propose as combination meta-model a SVM classifier. The SVM classifier is also trained with individual scores as input features. In Sect. 4 we will show the results for the three combination methods. Furthermore, we will discuss in-depth the results of the best performance model. Notice that by building a model, that takes into account browser context and behavioral dynamics scores, more confidence on the legitimacy of a given log-in attempt can be gained.

Scalability of the combined model Note that the models obtained for the two risk assessment strategies involve training with a dataset of multiple users, however one model is generated that can be applied for each user (there is no need to build one model for each user). Therefore, the approach is designed to scale to millions of users, once the two respective models are trained.

4 Evaluation

In order to train and evaluate the performance of our proposed method we collect two sets of data. The behavioral data set, containing both mouse and keyboard data, was retrieved from a public data set known as *The Wolf Of SUTD (TWOS)* [12] as we described in Sect. 2.3. Conversely, the context analytics data set was collected in house from

banking web services. This data set contains information about context-based features for online banking log-in sessions. The context-based data set has ca. 13 million entries summarizing connection features when users perform a password-based authentication process. Within those features each entry has information of session timestamp, IP Address and user agent. To avoid over-fitting we first split the history logs for each user into halves. The split is performed for both datasets. The reason of the last is to simulate an scenario where the user's credentials were compromised or guessed at some point. We use the first half to fit and validate the model. The second half is used to provide an unbiased evaluation of the final trained model.

In order to test the combined model we perform a match between the session attempts in TWOS data and context-based data. First we find out the data set with less entries, for us TWOS data. Afterwards we split the TWOS data set into positive (impersonation attacks) and negative samples. As we balanced TWOS dataset before we train the behavioral model the behavioral data has as many positives as negatives entries. We take the positives entries of TWOS and split them into two sets. One of those subsets is matched with an equal number of random sessions from the context-based data set. In that vein, the remaining subset is matched with negatives samples from context-based data. The same process is performed for the positive entries in TWOS data set. As a result, the data set for the combined model is distributed as Table 3 shows.

4.1 Session context-based model

Historically, the analysis of connection features is the most common technique to mitigate the risk of impersonation attacks. For this, we first train a model for the context-based information to have a notion of how the basic model is performing. Starting from the session timestamp, IP Address and user agent in the session start we calculate the convolutions and probability profiles described in Sect. 3. From the ca. 13 million session log in attempts, we take the 30% of data to test the models performance and the remaining to train algorithm. The model used to predict the risk of a connection based on contextual information was a random

Table 3 Distribution of combined label to test the model that aggregates the predictions of single models

Label behavioral	Label context	Data percentage (%)	Combined label
0	0	25	0
0	1	25	1
1	0	25	1
1	1	25	1

forest. The evaluation of the performance is done using standard classification evaluation measures. Using a confusion matrix, the following measures are calculated:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = \frac{2P\Delta R}{P + R}$$

where P , R , TP , FN , TN and FP are the precision score, recall score, the numbers of true positives, false negatives, true negatives and false positives, respectively. To evaluate the context-based model we define as positive the sessions with context simulation. The sessions with no context simulation attacks are the negative ones. As we are facing a classification problem some performance metrics are dependent of the decision threshold λ . The λ parameter defines the minimum output probability a prediction must hold to be

classified as a attack. Table 4 summarizes the performance of the single random forest model trained to alert attacks based on device context information, assuming the correct label for each sample is only the context simulation label.

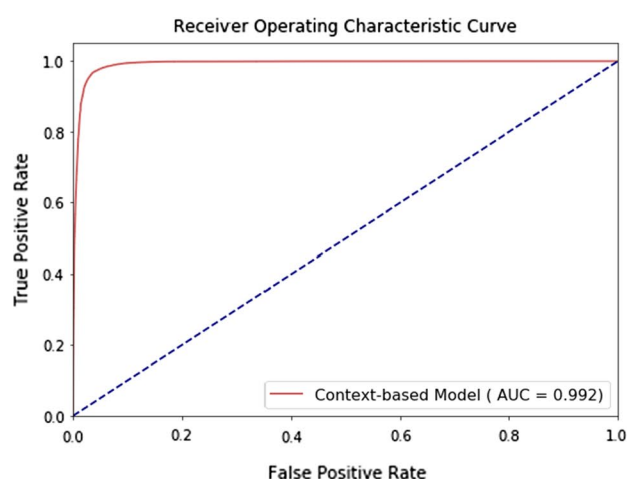
The results in Table 4 show the context-based model performs impressively. Particularly, we observe that the model has a high accuracy in the process of detect impersonation attacks with not large decreases in precision or recall. To have a better understating on how the model is performing depending on the threshold we compare precision and recall curves Fig. 5b for the context-based model. Furthermore, we show in Fig. 5a the model area under the curve (AUC) metric.

However, the results showed in Table 4 hide that in real scenarios an attack could be an impersonation of context (browser type, country of origin of IP etc.) or an impersonation of behavioral features for a given user at login time. Additionally, the imitation or simulation of the session context by an attacker is relatively easy. On account of this premise it is important to evaluate the model using as correct output for each sample the combined label. In other words, we define as positive the sessions with context simulation or impersonation attacks. The sessions without any attack are the negative ones. In the Table 5 we summarize the performance of the single context-based model for different decision thresholds when the combined label is the output variable.

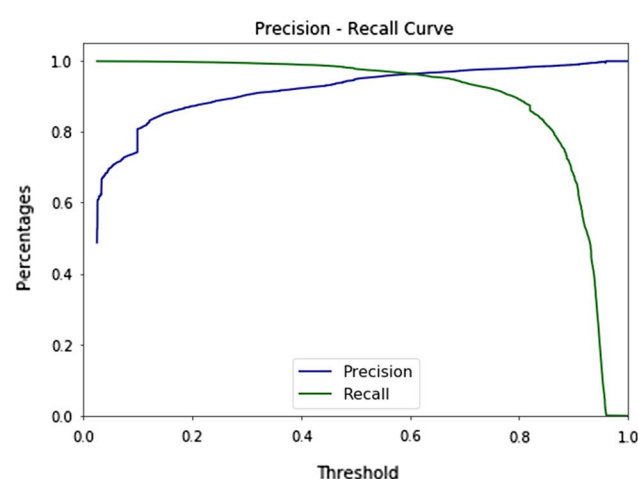
It becomes noticeable in Table 5 that context-based model is not resilient to complex attacks in which the attacker carefully manipulates HTTP parameters. Specifically, we observe that the model has decreased accuracy metric in almost 17% when the model prediction is evaluated over the combined label. To have a better understating on how

Table 4 Single context-based model performance for different classification thresholds when evaluated against context-session attacks

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.931	0.872	0.928	0.998
0.4	0.956	0.924	0.955	0.990
0.5	0.963	0.949	0.964	0.978
0.6	0.964	0.964	0.965	0.965
0.8	0.934	0.982	0.939	0.892



(a) ROC curve and AUC score.



(b) Precision-Recall curve.

Fig. 5 Threshold dependent performance curves for the single model of context-based analysis of HTTP connections when evaluated against context-session attacks

Table 5 Single context-based model performance for different classification thresholds when evaluated against session and biometric attacks

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.806	0.932	0.751	0.710
0.4	0.799	0.958	0.748	0.685
0.5	0.792	0.972	0.743	0.668
0.6	0.784	0.980	0.738	0.654
0.8	0.747	0.990	0.703	0.599

the model is performing depending on the threshold we compare precision and recall curves Fig. 6b for the context-based model. Furthermore, we show in Fig. 6a the model area under the curve (AUC) metric.

4.2 Behavioral biometrics model

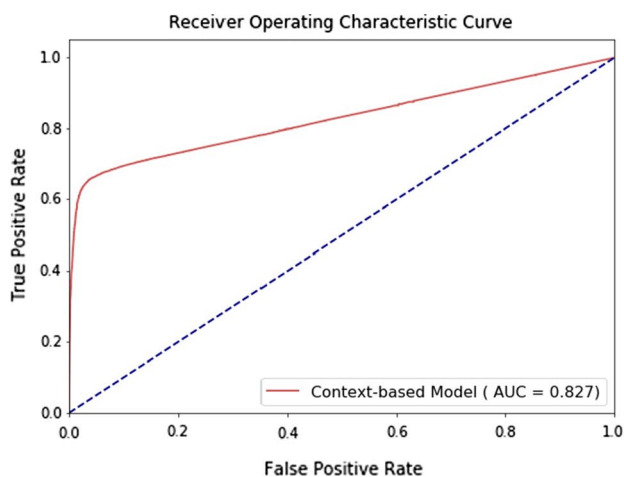
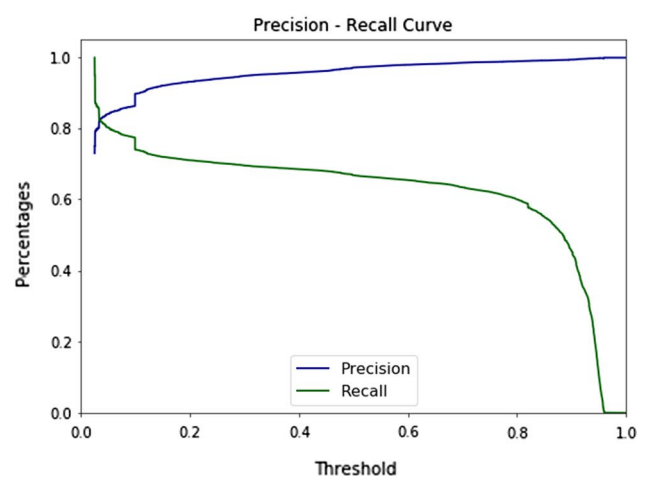
For the behavioral dynamics analysis, we extract mouse traces and keystrokes from the TWOS dataset for all users. After extraction we correlate the mouse and keyboard data for the different sessions the users performed. For each user, a session is created by collecting all mouse entries within a time window before the final login click is observed. Afterwards, the keyboard data is correlated searching for all keystrokes in keyboard data set within the same time windows for the same user. Only sessions with both mouse and keyboard information are considered, the others were ignored. With both mouse and keyboard session's information, we created the features described in Sect. 3. Once the feature data sets are correlated we train the behavioral dynamics model. A random forest was trained in order to capture the

Table 6 Single behavioral model performance for different classification thresholds (λ) when evaluated only against behavioral attacks

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.806	0.932	0.751	0.710
0.4	0.799	0.958	0.748	0.685
0.5	0.792	0.972	0.743	0.668
0.6	0.784	0.980	0.738	0.654
0.8	0.747	0.990	0.703	0.599

behavioral patterns of each user. In order to test the performance of the model we split the half data set into 70% to train and the remaining data's entries to validate model. The evaluation of the performance is done using standard classification evaluation measures explained in Sect. 4. We test the model with the second half of the behavioral data. First, we define as positive(attacks) the sessions with only impersonation attacks. The sessions without impersonation attacks are the negative ones. In the Table 6 we summarize the performance of the single behavioral dynamics model for different decision threshold, assuming the correct label for each sample is the behavioral biometrics label.

From the Table 6 we observe the model has a high precision for higher thresholds. However, more the threshold is increased more the recall decrease drastically. As result, the behavioral biometrics model exhibits a high rate of false negatives when high thresholds are required. Notice that the problem addressed in this paper considers the high cost of false negative predictions because they generates a cascade of attacks which the system does not alert. For this reason we compare precision and recall curves Fig. 7b for the

**(a)** ROC curve and AUC score.**(b)** Precision-Recall curve.**Fig. 6** Threshold dependent performance curves for the single model of context-based analysis of HTTP connections when evaluated over attacks to context and biometrics

behavioral model to find out the threshold which minimizes the critical cases. Additionally, we show in Fig. 7a the model receiver operating characteristic(ROC) curve performance.

The evaluation using only the behavioral label gives that the threshold which minimizes the false positives and the false negatives is close to 0.2. But these results have a bias because of complex attacks could arise from behavioral biometrics impersonation but also from session context simulation. As we did with the session context-based model, we evaluate the model using as output variable the combined label. For that purpose, we define as positive the sessions with context simulation or impersonation attacks. The sessions without any attack are the negative ones.

The results for the combined label in Table 7 show that the model keeps high precision independent of the decision threshold. However, the false negative rate increases considerably. Furthermore, the F1-score is 14% lower which predicts a decrease in AUC metrics. To corroborate the decrease in AUC we show in Fig. 8a the model receiver operating characteristic(ROC) curve performance. Additionally, it is presented next the precision and recall curves Fig. 8b for the behavioral model.

The AUC scores for both models are around 0.80, however, the precision and the recall metrics are not accurate enough for the problem we are addressing.

For instance, the recall for context based model indicates a high rate of false negatives which in our context means a high rate of attacks are unnoticeable for the system. Moreover, F1-scores denote that each model separately has a similar performance when they try to detect a global attack. The issue is therefore that each model is not able to detect the counterpart attack: the context-based model will not detect

Table 7 Single behavioral model performance for different classification thresholds (λ) when evaluated over the combined label

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.800	0.837	0.720	0.765
0.4	0.662	0.896	0.608	0.525
0.5	0.543	0.920	0.526	0.385
0.6	0.419	0.940	0.454	0.269
0.8	0.222	0.970	0.359	0.126

changes in biometric features, and the behavioral model, on its own, will ignore changes in the connection context.

4.3 Combined model

In the light of the above results we develop a model that attempts to overcome the shortcomings of the single risk assessment strategies (context-based analysis of HTTP connections and behavioral dynamics individually) by proposing a single model that takes into account both strategies. In the following we show the results for each combination model we proposed in Sect. 3.3: (1) Parametric Linear Combination, (2) Random Forest classifier and (3) Support-Vector Machine classifier.

4.3.1 Parametric linear combination

For the convex linear model we combine the predictions of single models using Eq. 3. The values showed in this section were calculated using $\alpha_c = 0.5$ and $\alpha_b = 0.5$ following the intuition that both attacks are equally probable in our data

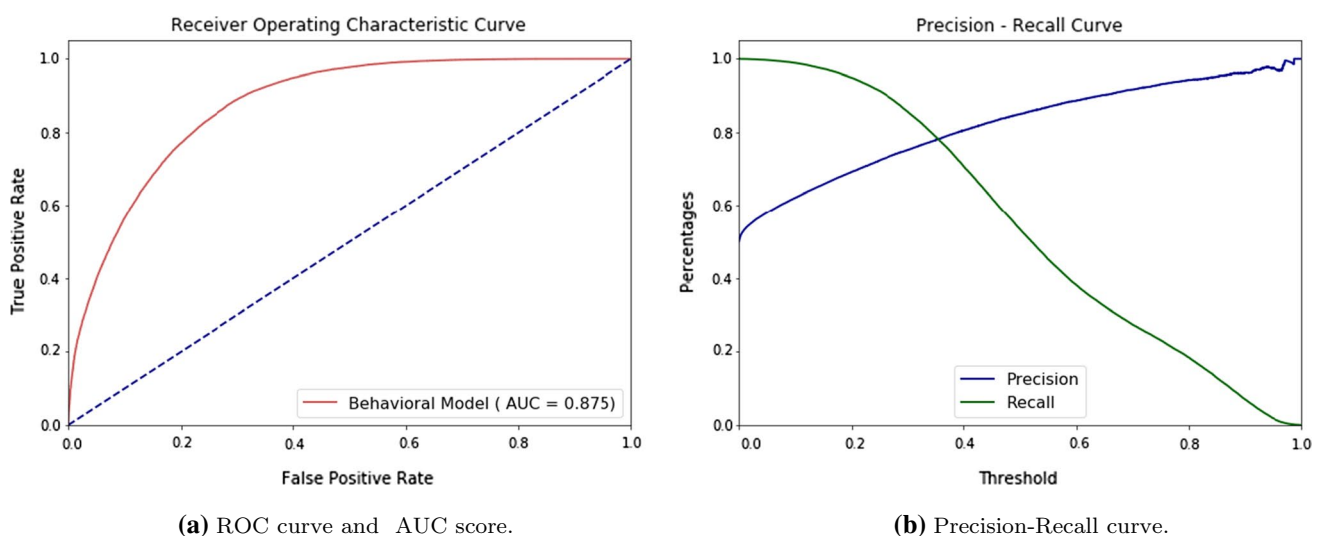


Fig. 7 Threshold dependent performance curves for the single model of behavioral dynamics when evaluated against behavioral attacks

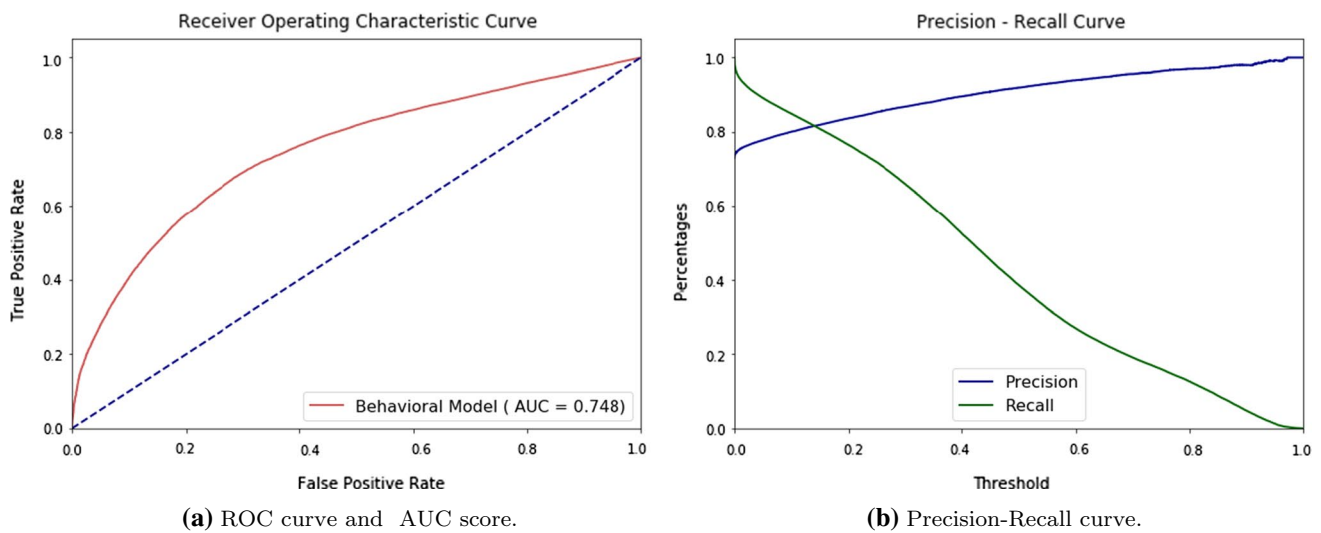


Fig. 8 Threshold dependent performance curves for the single model of behavioral dynamics when evaluated over the combined label ($\alpha_b = \alpha_c = 0.5$)

Table 8 Combined model performance for different classification thresholds when evaluated over the combined label ($\alpha_b = \alpha_c = 0.5$)

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.915	0.894	0.873	0.937
0.4	0.843	0.972	0.797	0.744
0.5	0.699	0.986	0.660	0.542
0.6	0.549	0.992	0.545	0.379
0.8	0.185	0.998	0.344	0.102

set construction. However, those parameters for the convex combination were optimized at the end of the entitled subsection. Table 8 shows the results for the combined model for different classification thresholds.

It becomes noticeable that the parametric linear combination model have a considerable effect in the improvement accuracy of detect impersonation attacks. A detailed comparison of the single behavioral model, the single context-based model and the combined model is presented in Table 9.

The results achieved with the parametric linear combination model show an important enhancement in detection of attacks, as Table 9 reveals. The high precision and recall values bring to light that the use of a combined model performs better in detecting attacks, given that the combined model can recognize both changes in context and changes in behavior. At the same time, an improved F1-Score and accuracy show that the overall classification was improved, thus also false positives caused by use of new devices or travel can be sometimes mitigated by using the information from the behavioral model (Fig. 9).

Table 9 Model performance comparison with a decision threshold of 0.2 for the three models we build to increase security in login attempts when evaluated over the combined label

Model	F1-score	Precision	Accuracy	Recall
Behavioral	0.800	0.837	0.720	0.765
Context-based	0.806	0.932	0.751	0.710
Parametric linear combination	0.915	0.894	0.873	0.937

Finally, we show in Fig. 10 the receiver operating characteristic (ROC) curve for all models we discuss in this work. As it is also evident from the AUC in this figure, a combined model using the parametric linear combination has better performance than the individual models in the data set we have considered.

Up to this point all the results we have presented are related to the scenario when $\alpha_c = 0.5$ and $\alpha_b = 0.5$. Despite the fact we chosen those values based on the intuition of that both attacks are equally probable in our data set construction, there might more optimal parameters. To find the best set of parameters we performed an exhaustive search of the weights in the convex linear combination. In Table 10 we present the F1-Score, Precision, Accuracy and Recall for 20 different configurations of parameters with a threshold $\lambda = 0.2$. Remember the linear combination in Eq. 3 is convex and the parameters must satisfy that $\alpha_c + \alpha_b = 1$. Some of the configurations are presented in Fig. 11 in a graphical way to have a better understanding of the model behavior when the α_b is modified.

The information in Table 10 show that the behavioral model should have a greater contribution in the linear

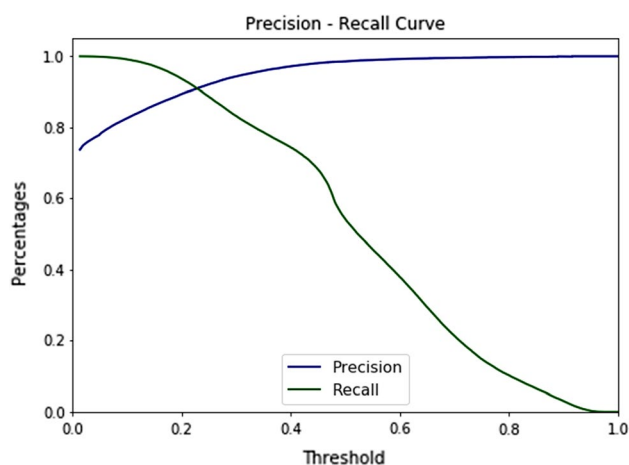
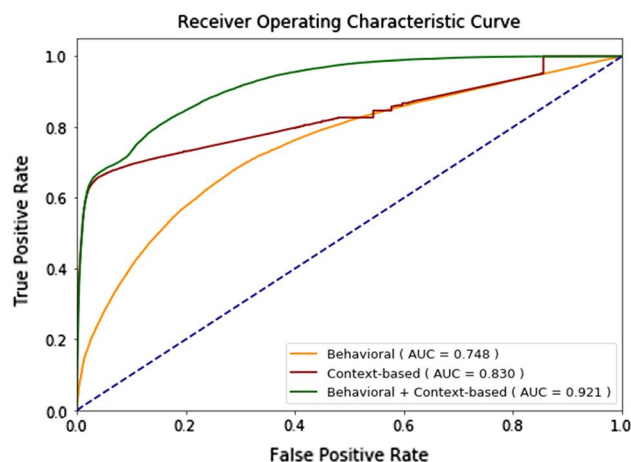
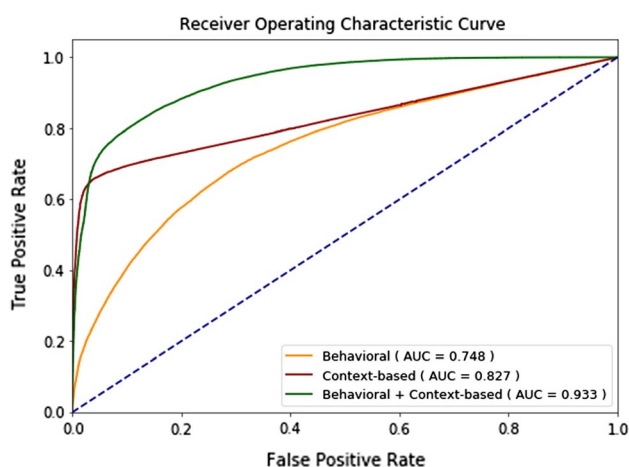
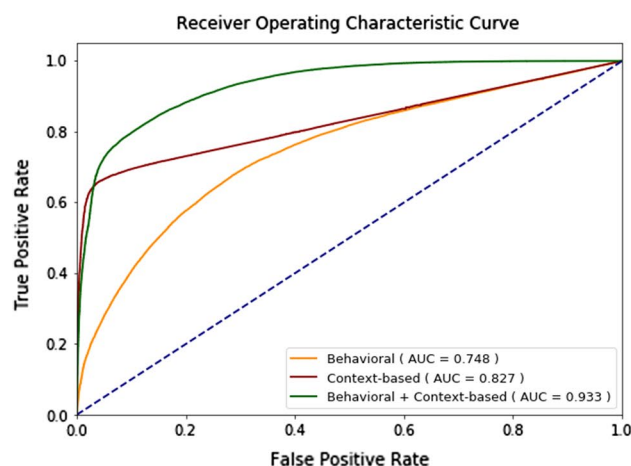


Fig. 9 Precision - Recall curves for the combined risk assessment model (i.e. context-based analysis of HTTP connections and behavioral dynamics) using $\alpha_b = \alpha_c = 0.5$



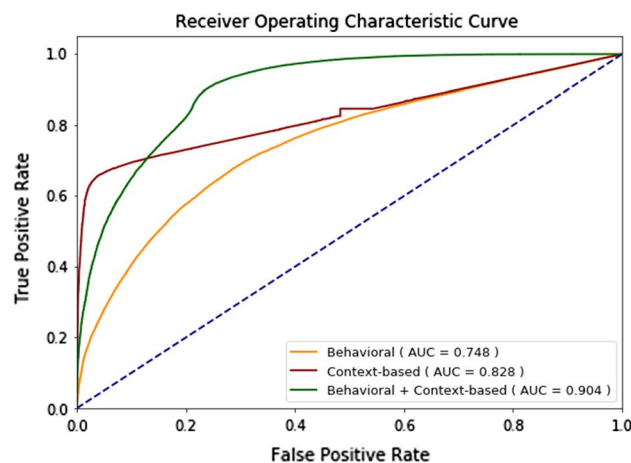
(a) Model's ROC curve and AUC score for $\alpha_b = 0.3$.



(b) Model's ROC curve and AUC score for $\alpha_b = 0.5$.

Fig. 10 ROC curves and AUC scores for the single risk assessment strategies(context-based analysis of HTTP connections and behavioral dynamics) and the **Parametric Linear Combination Model** which combines both strategies (using $\alpha_b = \alpha_c = 0.5$.)

combination because it maximises three out of four metrics we evaluated in the test dataset. As it can be seen from the analysis, when the parameter α_b is set to 0.70 the F1-Score, Precision and Recall are maximized, while the accuracy has a decrease of 6% compared to the maximum value. It is important to remark that one the major objectives of the combined model was to reduce the false negative and false positive rates and choice of $\alpha_b = 0.70$ succeed this goal, while it keeps a competitive accuracy. Moreover, the results of the exhaustive analysis we performed mean that changes in biometrics features are more difficult to detect compared to changes in session context because of the need of increase the contribution of behavioral model to maximize the combined model performance.



(c) Model's ROC curve and AUC score for $\alpha_b = 0.7$.

Fig. 11 ROC curves and AUC scores for the single risk assessment strategies(context-based analysis of HTTP connections and behavioral dynamics) and the model which combines both strategies for different set-ups of α_b

Table 10 Parametric linear combination model performance comparison for different set-ups of parameters α_b and α_c

Parameter α_b	F1-score	Precision	Accuracy	Recall
0.00	0.751	0.806	0.932	0.710
0.05	0.753	0.808	0.932	0.714
0.10	0.756	0.811	0.932	0.718
0.15	0.766	0.820	0.932	0.733
0.20	0.787	0.840	0.931	0.766
0.25	0.816	0.866	0.927	0.812
0.30	0.834	0.881	0.921	0.845
0.35	0.850	0.895	0.914	0.877
0.40	0.861	0.904	0.907	0.902
0.45	0.869	0.911	0.900	0.922
0.50	0.873	0.915	0.894	0.937
0.55	0.875	0.917	0.888	0.948
0.60	0.876	0.918	0.884	0.955
0.65	0.876	0.919	0.881	0.960
0.70	0.876	0.919	0.878	0.963
0.75	0.873	0.917	0.876	0.962
0.80	0.823	0.881	0.866	0.896
0.85	0.785	0.852	0.857	0.847
0.90	0.760	0.833	0.850	0.816
0.95	0.739	0.815	0.843	0.789

Only α_b is presented due to the linear combination is convex and thus parameters must satisfy that $\alpha_c + \alpha_b = 1$

Bold values highlight the value of α_a for which the corresponding metric (F1-score, Precision, Accuracy, Recall) is maximized

4.3.2 Random forest classifier

For the Random Forest meta-model we feed a random forest classifier with both scores given by single models: context-based and behavioral score. Due to the dimensionality of input vector we propose to train a random forest with 10 trees. Table 11 shows the results for the random forest meta-model for different classification thresholds.

Table 11 let us to conclude that best classification threshold for the random forest meta-model is close to 0.2. Moreover, Precision metric obtained by using the random forest is higher than using the parametric linear combination, while the Recall metrics is lower. A comparison of random forest meta-model performance metrics compared to the single behavioral model and the single context-based model is presented in Table 12.

The use of a random forest classifier as meta-model improves the detection of combined attacks, as results in Table 12 reveal. The high F1-Score and accuracy values show that overall classification of attacks was improved in relation to the single models. In detail, the random forest approach exhibits a good performance to avoid false negatives due to high Precision value. At the same time, the accuracy and Recall metrics are lower compared to linear

Table 11 Random forest model performance for different classification thresholds when evaluated over the combined label

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.905	0.914	0.863	0.897
0.4	0.897	0.929	0.855	0.868
0.5	0.892	0.935	0.850	0.854
0.6	0.885	0.940	0.842	0.837
0.8	0.860	0.953	0.813	0.783

parametric meta-model. Finally, we show in Fig. 12b the receiver operating characteristic (ROC) curve for the single models and the random forest meta-model. As it is also evident from the AUC in this figure, a combined model using the random forest classifier has better performance than the individual models in the data set we have considered.

4.3.3 Support-vector machine classifier

For the Support-Vector Machine meta-model we feed a SVM classifier with both scores given by single models: context-based and behavioral score. As we interested on to give a risk assessment of the incoming login we have to calibrate typical SVM's class scores into probabilities. In order to have a probability score we use logistic regression on the SVM's scores, fit by an additional cross-validation on the training data. Table 13 shows the results for the SVM meta-model for different classification thresholds.

The results achieved by using the SVM classifier as the metal model show a important improvement in the detection impersonation attacks compared to single models. It is interesting that the SVM meta-model exhibits an almost constant, and high, value for all metrics evaluated no matter the selected decision threshold. A detailed comparison of performance for the single behavioral model, the single context-based model and the combined model using the SVM classifier is presented in Table 9.

The information in Table 14 show that SVM meta-model approach is also capable to detect with high accuracy the impersonation attacks. The SVM classifier exhibits the best behavior to minimize false positives and false negatives

Table 12 Random forest combination model performance compared with the best classification decision threshold for the single models when evaluated over the combined label

Model	F1-score	Precision	Accuracy	Recall
Behavioral	0.800	0.837	0.720	0.765
Context-based	0.806	0.932	0.751	0.710
Random Forest Meta-Model	0.905	0.914	0.863	0.897

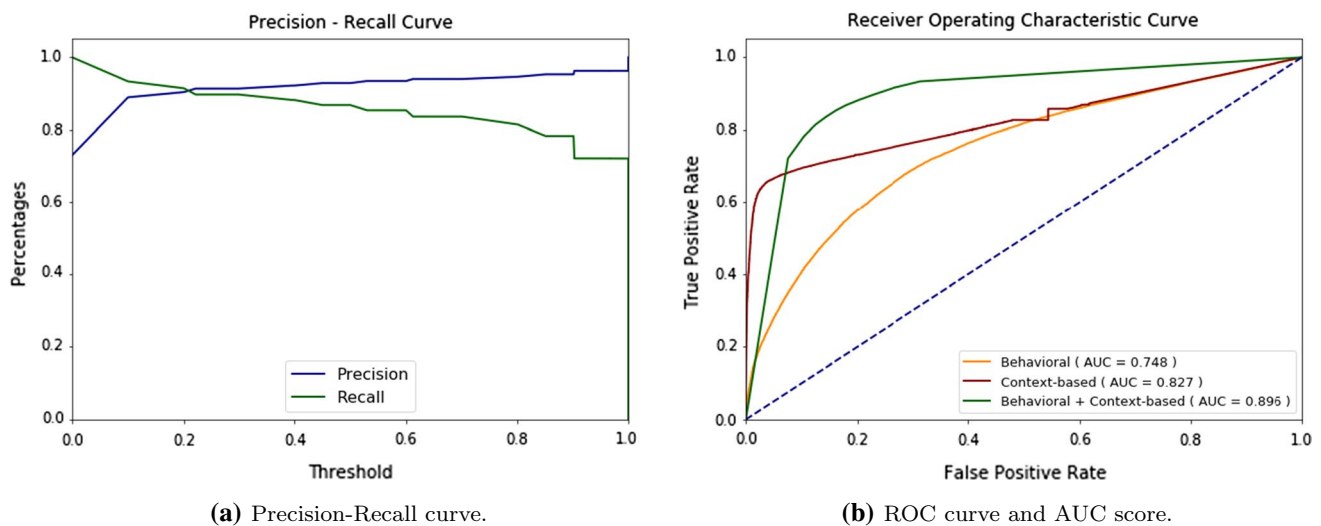


Fig. 12 Threshold dependent performance curves for combined model using **Random Forest Classifier** as meta-model

Table 13 SVM classifier model performance for different classification thresholds when evaluated over the combined label

Decision threshold	F1-score	Precision	Accuracy	Recall
0.2	0.914	0.915	0.874	0.912
0.4	0.908	0.926	0.869	0.892
0.5	0.906	0.930	0.866	0.883
0.6	0.904	0.934	0.864	0.875
0.8	0.895	0.943	0.854	0.852

at the same time, as precision and recall metrics reveals. Moreover, the F1-Score and accuracy show high rates of attacks detection for the combined label. Finally, we depict in Fig. 13b the receiver operating characteristic (ROC) curve for the single models and the SVM meta-model.

Notice that at the optimal decision threshold the results for the SVM meta-model indicate the best trade-off for all four metrics evaluated in our work. The SVM exhibits the best behavior of the three approaches to minimize the number of false positives and false negatives. Moreover, the SVM meta-model has the highest values of Precision and Accuracy. Finally, the F1-score obtained for SVM classifier is statistically comparable with the best F1-Score obtained for parametric linear combination.

Scalability of the combined model We have measure the time it takes to evaluate a given session against the separate strategies, in order to assess the scalability of the approach. These times were measured in a i7-7700hq processor (2.8 ghz), using a single core. For the context-based model, we obtain an execution time of 105 ms in average ($\pm 435 \mu\text{s}$) per session. For the behavioral dynamics model we can classify a session within 106 ms ($\pm 263 \mu\text{s}$) per session. The times

Table 14 SVM combination meta-model performance compared with the best classification decision threshold for the single models when evaluated over the combined label

Model	F1-score	Precision	Accuracy	Recall
Behavioral	0.800	0.837	0.720	0.765
Context-based	0.806	0.932	0.751	0.710
SVM Meta-model	0.914	0.915	0.874	0.912

related to the combination method depends on the combination approach: for the (1) Parametric Linear Combination we obtained an execution time of 27 ns in average ($\pm 0.8 \text{ ns}$), for the (2) Random Forest meta-model we obtained an execution time of 111 ms in average ($\pm 814 \mu\text{s}$) and for the (3) SVM classifier we obtained an execution time of 1.1 ms in average ($\pm 22 \mu\text{s}$). As a result, the risk-assessment can be completed within half second per each session.

4.4 Use in industrial scenarios

There is no single approach for including an anomaly detection system such as the one discussed in this paper in an operational workflow, nor a one-size-fits-all choice of parameters. Smaller entities – especially if not experiencing a high level of fraud – may want to handle assessments manually. In this case human operators in a SOC receive an alert and react to it. Actions may include blocking the user account, or contacting the user. Aggregated data can also be used to drive the decision process towards more sophisticated, and effective, solutions.

In this scenario where all alerts are handled by a human operator it is mandatory that the alert rate is reasonably

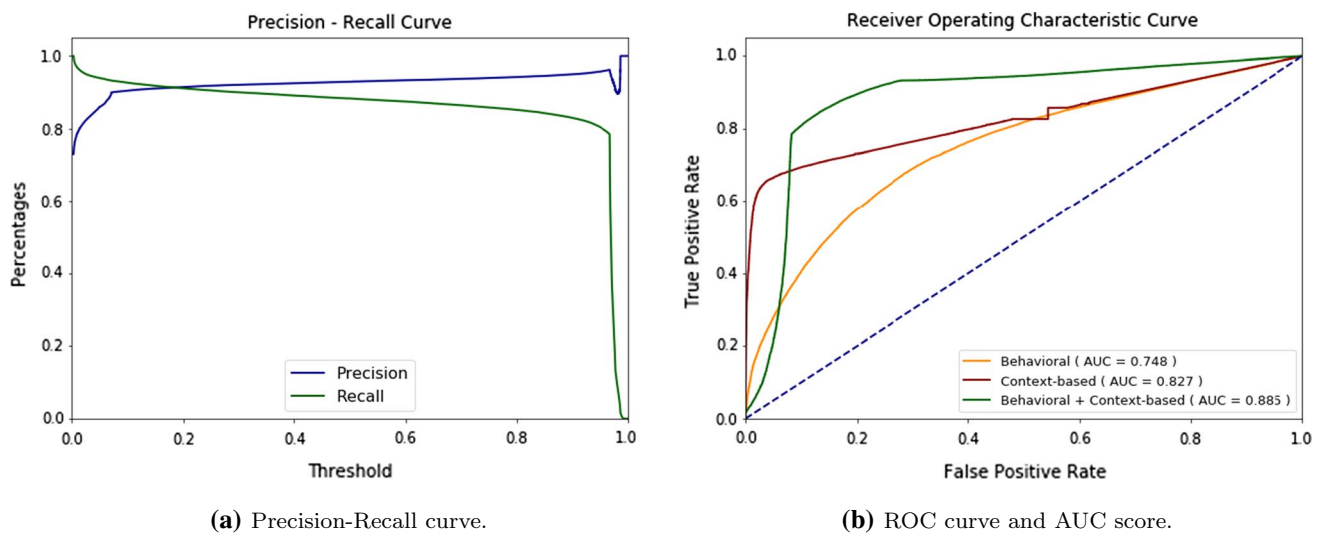


Fig. 13 Threshold dependent performance curves for combined model using **SVM Classifier** as meta-model

small. While it is impossible to define a generic threshold, 1% may be a useful benchmark. As the user base, or the fraud level, grows the institution may decide to integrate the assessments in the application work-flow. If this is done with hard-coded rules then a low rate of false positives is critical, as each alert will generate an action and therefore an expense.

Bigger, or more security-aware, institutions will probably feed the assessment generated from this module to additional systems. While in practice there is no such clear distinction – one single system can often play the two roles – we can typify this systems in two categories:

- *Dynamic authentication systems.* Based on the assessment and other factors such as the user's risk profile and the money at stake the system can decide if additional authentication factors are to be requested to the user, or if access has to be blocked altogether, once or permanently.
- *Transaction anomaly detection systems,* that can include the information related to the transaction to decide if it can be approved, denied or further action should be requested, including sending the transaction to a SOC for further human analysis.

In this last scenario a higher rate of false positives is acceptable, as the alerts will be filtered using independent criteria. Furthermore, a numeric assessment is preferable with respect to a binary value, letting the institution fine tune, possibly in real time, how to react to the assessment.

In the context of web-application static authentication, we believe that optimizing the choice of parameters in the model to minimize false negatives (i.e. undetected attacks), is acceptable if in those cases, users can be prompted for a

2-Factor-Authentication such as an OTP sent to their mobile phones. In our model, setting the threshold between 0.2 and 0.25 will yield between 35% and 24.6% false-positive rate against 1.9% and 3.5% false-negative rate respectively. This means roughly one out four users is prompted for 2-FA, whereas between one out 30 to 50 attacks goes undetected. Note that these number hold for our experiments, where 75% of the data consists of attacks, in practice attacks are much less common.

For very sensitive customers, further manual action can be taken depending on the transactions performed in the application. For instance, in the banking domain, further filters depending on transaction amounts can be applied, given a suspicion on context and behavior.

4.5 Discussion and limitations

We have shown that in principle the combination of both risk-based authentication strategies indeed improves the performance of the single models in isolation. There are a number of limitations to our evaluation. First, the data from the HTTP contextual model and the behavioral model do not belong to the same users. Although in principle there should be no strong correlation between context and behavior, a more accurate model could be built if variations in behavior from the same user across devices are taken into account.

On the other hand, experiments were built under the assumption that the combinations of different scenarios (between contextual and behavioral attacks) were equally likely. In practice, attacks are rare, and this aspect should be considered in future work. Last, we have considered behavioral data that has been adapted to simulate static authentication, but that in reality may belong to other activities

in the context of the competition where it was gathered. In future work, we plan to consider data collected from real user log-ins. To the best of our knowledge, there is no public database containing both mouse and keyboard data for static authentication, although there are some datasets containing either of them.

5 Related work

Risk-based authentication has seen popularity in web applications due to the limitations of password authentication. Bonneau et al. [2] give a historical overview of the introduction of risk-based authentication in practical systems in order to complement password-based authentication. Alaca and Van Oorschot [5] classify and survey several device fingerprinting mechanisms that can be used as the basis for authentication, and discuss different ways in which authentication can be complemented by them. Misbahuddin et al. [20] study the application of machine learning techniques for risk-based authentication using HTTP and network patterns, in a similar spirit of our technique, but do not take into account behavioral biometric patterns from mouse and keyboard, that as we have shown, improve the accuracy of risk-based authentication.

On the other hand, there are several works exploring applications of behavioral biometrics for static and continuous authentication. In the general context of desktop based applications, Mondal and Bours [9] have studied the combination of keyboard and mouse for continuous authentication. Different from them, we focus on static authentication for web applications. Shen et al. [10] study the applicability of mouse-based analytics for static authentication and conclude that longer than typical log-in interactions would be necessary in order to obtain high accuracy in such models. Traore et al. [13] explore the combination of both mouse and keyboard for risk-based authentication in web applications, however they assume the behavior monitor to be in the application after log-in as well (continuous authentication), and obtain an equal error rate of around 8% (even when considering full web sessions). Recently, Solano et al. [21] study the use of mouse and keyboard features for static authentication in web applications, however they focus the research on the feasibility of learning user behavior by using only few samples from the legitimate user in the training phase. Moreover they do not consider other risk factors (such as context) in their approach.

To the best of our knowledge the combination of traditional risk-based authentication based on HTTP and network information and behavioral biometrics for static (log-in time) authentication, as proposed in this work, has not been discussed in the literature.

6 Conclusions

The results of our proposed method demonstrates that device identification and behavioral analytics are complementary methods of risk measurement thus by combining both of them, efficacy and performance are never lower than single method approach. Moreover, our approach appears to be more resilient to changes, for instance when a user changes his/her device, an only device identification approach will alert event though there is no attack and an only behavioral approach will not notice the change at all.

In this work we also have shown that, by combining both device identification and behavioral identification risk assessment methods during login time, static web authentication performance can be enhanced by detecting single and mixed attack models with higher or equal accuracy in each case. This also makes web authentication systems more robust and may give the user a better security experience.

We have also discussed the practical applicability of our solution in industrial scenarios. In the future, we plan to consider a more powerful attacker model that is aware of a behavioral risk assessment component and attempts to bypass it, as well as reproducing this experiments on novel datasets that collect both session information and behavioral dynamics simultaneously.

Compliance with ethical standards

Conflict of interest All authors were Cyxtera employees (now AppGate Inc.) at the time of writing this manuscript and declare no conflict of interest. Parts of this study use the TWOS dataset, which is a public dataset based on the behaviour of 24 students during a gamified experiment and shared in an anonymized fashion by the Singapore University of Technology and Design. Authors of the original study obtained SUTD's IRB consent to carry out and share the data used in this paper.

Ethical standard In this work we also used a proprietary dataset of log-in contextual information (based on HTTP parameters), that was anonymized and which cannot be associated with any particular individual. Moreover, we only disclose aggregated results based on this dataset. So in sum all procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

References

1. Perrig, A.: Shortcomings of password-based authentication. In: 9th USENIX Security Symposium, vol. 130. ACM (2000)
2. Bonneau, J., Herley, C., Stajano, F.M., et al.: Passwords and the evolution of imperfect authentication. *Commun. ACM* **58**, 78–87 (2014)
3. Newman, L.: Hacker Lexicon: What is Credential Stuffing? *Wired Magazine* (2019). <https://www.wired.com/story/what-is-credential-stuffing/>. Accessed 12 Sept 2019

4. Kaspersky: Zeus malware. Online (2019). <https://usa.kaspersky.com/resource-center/threats/zeus-virus>. Accessed 12 Sept 2019
5. Alaca, F., Van Oorschot, P.C.: Device fingerprinting for augmenting web authentication: classification and analysis of methods. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 289–301. ACM (2016)
6. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In: Stolfo, S.J., Bellovin, S.M., Keromytis, A.D., Hershkop, S., Smith, S.W., Sinclair, S. (eds.) *Insider Attack and Cyber Security*, pp. 69–90. Springer, Boston (2008)
7. Yampolskiy, R.V., Govindaraju, V.: Behavioural biometrics: a survey and classification. *Int. J. Biom.* **1**(1), 81–113 (2008)
8. Zheng, N., Paloski, A., Wang, H.: An efficient user verification system via mouse movements. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 139–150. ACM (2011)
9. Mondal, S., Bours, P.: Combining keystroke and mouse dynamics for continuous user authentication and identification. In: 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), pp. 1–8. IEEE (2016)
10. Shen, C., Cai, Z., Guan, X., Wang, J.: On the effectiveness and applicability of mouse dynamics biometric for static authentication: a benchmark study. In: 2012 5th IAPR International Conference on Biometrics (ICB) (2012)
11. Solano, J., Camacho, L., Correa, A., Deiro, C., Vargas, J., Ochoa, M.: Risk-based static authentication in web applications with behavioral biometrics and session context analytics. In: Zhou, J., Deng, R., Li, Z., Majumdar, S., Meng, W., Wang, L., Zhang, K. (eds.) *Applied Cryptography and Network Security Workshops*, pp. 3–23. Springer, Berlin (2019)
12. Harilal, A., Toffalini, F., Homoliak, I., Castellanos, J., Guarnizo, J., Mondal, S., Ochoa, M.: The wolf of SUTD (twos): a dataset of malicious insider threat behavior based on a gamified competition. *J. Wirel. Mob. Netw.* (2018). <https://doi.org/10.22667/JOWUA.2018.03.31.054>
13. Traore, I., Woungang, I., Obaidat, M.S., Nakkabi, Y., Lai, I.: Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In: 2012 Fourth International Conference on Digital Home (2012)
14. Swati Gurav, R.G., Mhangore, S.: Combining keystroke and mouse dynamics for user authentication. *Int. J. Emerg. Trends Technol. Comput. Sci. (IJETTCS)* **6**, 055–058 (2017)
15. Cao, Y., Li, S., Wijmans, E.: (Cross-)browser fingerprinting via OS and hardware level features. In: NDSS (2017). <https://doi.org/10.14722/ndss.2017.23152>
16. Nakibly, G., Shelef, G., Yudilevich, S.: Hardware fingerprinting using HTML5 (2015). [arXiv:1503.01408v3](https://arxiv.org/abs/1503.01408v3)
17. Sanchez-Rola, I., Santos, I., Balzarotti, D.: Clock around the clock: time-based device fingerprinting. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1502–1514 (2018)
18. Kohno, T., Broido, A., Claffy, K.C.: Remote physical device fingerprinting. *IEEE Trans. Dependable Secure Comput.* **2**(2), 93–108 (2005)
19. Bailey, K.O., Okolica, J.S., Peterson, G.L.: User identification and authentication using multi-modal behavioral biometrics. *Comput. Secur.* **43**, 77–89 (2014)
20. Misbahuddin, M., Bindhumadhava, B.S., Dheeptha, B.: Design of a risk based authentication system using machine learning techniques. In: 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation, pp. 1–6 (2017)
21. Solano, J., Tengana, L., Castelblanco, A., Rivera, E., Lopez, C., Ochoa, M.: A few-shot practical behavioral biometrics model for login authentication in web applications. In: NDSS Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb'20) (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.