DISSERTAÇÃO DE MESTRADO PROFISSIONAL

AUTENTICAÇÃO CONTÍNUA BASEADA EM BIOMETRIA COMPORTAMENTAL PARA APLICAÇÕES BANCÁRIAS *MOBILE*

Priscila Morais Argôlo Bonfim Estrela

Brasília, agosto de 2020

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CONTINUOUS AUTHENTICATION BASED ON BEHAVIORAL BIOMETRICS FOR MOBILE BANKING APPLICATIONS

AUTENTICAÇÃO CONTÍNUA BASEADA EM BIOMETRIA COMPORTAMENTAL PARA APLICAÇÕES BANCÁRIAS *MOBILE*

PRISCILA MORAIS ARGÔLO BONFIM ESTRELA

ORIENTADOR: WILLIAM FERREIRA GIOZZA, DR.
COORIENTADOR: DINO MACEDO AMARAL, DR.

DISSERTAÇÃO DE MESTRADO PROFISSIONAL EM
ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPEE.MP.001
BRASÍLIA/DF: AGOSTO - 2020

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

AUTENTICAÇÃO CONTÍNUA BASEADA EM BIOMETRIA COMPORTAMENTAL PARA APLICAÇÕES BANCÁRIAS *MOBILE*

Priscila Morais Argôlo Bonfim Estrela

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. William Ferreira Giozza, Ph.D, FT/UnB _____
Orientador

Prof. Rafael Timóteo de Sousa Jr, Ph.D, FT/UnB _____
Examinador Interno

Prof. Flávio Elias Gomes de Deus, Ph.D, FT/UnB _____
Examinador interno

FICHA CATALOGRÁFICA

MORAIS ARGÔLO BONFIM ESTRELA, PRISCILA
AUTENTICAÇÃO CONTÍNUA BASEADA EM BIOMETRIA COMPORTAMENTAL PARA APLICAÇÕES BANCÁRIAS *MOBILE* [Distrito Federal] 2020.
xvi, 145 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2020).
Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Autenticação contínua
2. Biometria comportamental
3. *Mobile*
4. Aplicações bancárias

I. ENE/FT/UnB
II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ESTRELA, P. M. A. B. (2020). *AUTENTICAÇÃO CONTÍNUA BASEADA EM BIOMETRIA COMPORTAMENTAL PARA APLICAÇÕES BANCÁRIAS MOBILE*. Dissertação de Mestrado Profissional, Publicação: PPEE.MP.001, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 145 p.

Priscila Morais Argôlo Bonfim Estrela
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico este trabalho a Deus em primeiro lugar, que me sustentou e me capacitou para concluir essa dissertação. Ao meu marido que esteve a todo momento me apoiando, mesmo nas horas mais difíceis. Á minha familia e amigos, dos quais tive que me privar de estarmos juntos por causa da jornada do mestrado e do desenvolvimento deste trabalho.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que me capacitou e me possibilitou mais esta conquista.

Agradeço ao meu marido, que me apoiou durante todo o período de construção deste trabalho.

Um agradecimento especial para os professores, William Ferreira Giozza, Robson de Oliveira Albuquerque e Dino Macedo Amaral que me apoiaram durante essa jornada, doando o seu tempo e experiência para que este trabalho pudesse ser desenvolvido.

Muito obrigada a todos os meus colegas e amigos, que me auxiliaram doando o seu tempo e conhecimento durante o desenvolvimento da pesquisa.

Agradeço também à Universidade de Brasília, e a todos os professores que compõem o corpo docente do meu curso, pela qualidade do ensino oferecido e empenho desprendido durante todo o curso.

RESUMO

Título: Autenticação Contínua Baseada em Biometria Comportamental para Aplicações Bancárias *Mobile*
Autor: Priscila Morais Argôlo Bonfim Estrela
Orientador: William Ferreira Giozza, Dr.
Coorientador: Dino Macedo Amaral, Dr.
Programa de Pós-Graduação Profissional em Engenharia Elétrica – Área de Concentração em Segurança Cibernética
Brasília, 6 de agosto de 2020

Na maior parte das aplicações *mobile* o primeiro passo, antes da interação com as funcionalidades oferecidas, é a autenticação do usuário. Geralmente a autenticação somente é executada em tempo de *login*, ou quando da confirmação da transação, mas esta abordagem pode expor os usuários à fraudes ligadas ao roubo de credenciais. Uma solução para superar essa vulnerabilidade seria a adoção de uma metodologia de autenticação contínua. Neste trabalho é proposto um *framework* multimodal para autenticação contínua e implícita para aplicações bancárias *mobile*, baseado na biometria comportamental no momento de digitação da senha, em tempo de *login*, na interação via *touchscreen* com a aplicação, pós *login* e na localização capturada via GPS para geração de alertas de segurança, caso um impostor tente se passar por um usuário legítimo na utilização da aplicação.

O *framework*, proposto e validado durante o desenvolvimento deste trabalho, demonstrou resultados de F1 *Score*, média harmônica entre o *recall* e a precisão, entre 90,68% e 97,05%, e percentual de erros referentes a impostores aceitos e usuários legítimos rejeitados, *Equal Error Rate* (EER), entre 9,85% e 1,88% para verificação estática, *login*, e dinâmica, pós *login*, que apontam a viabilidade do uso desse sistema proposto como mais uma camada de segurança, se utilizado em conjunto com métodos convencionais como senha.

Palavras-chave: Autenticação contínua, Biometria comportamental, *Mobile*, Aplicações bancárias.

ABSTRACT

Title: Continous Authentication Based on Behavioral Biometrics for Mobile Banking Applications
Author: Priscila Morais Argôlo Bonfim Estrela
Supervisor: William Ferreira Giozza, Dr.
Co-Supervisor: Dino Macedo Amaral, Dr.
Professional Post-Graduate Program in Electrical Engineering – Cybersecurity Concentration Area
Brasília, August 6, 2020

In most mobile applications the first step, before interacting with the offered features, is user authentication. Authentication is usually only performed at login time, or when the transaction is confirmed, but this approach can exposes users to fraud related to theft of credentials. One solution to overcome this vulnerability is continuous authentication. This work proposes a multimodal framework for continuous and implicit authentication in mobile banking applications, based on behavioral biometrics at the time of password typing, at login time, on touchscreen interaction with the application, post login and at location captured via GPS to generate security alerts if an impostor tries to impersonate a legitimate user when using the application.

The framework, proposed and validated during the development of this work, showed F1 Score, harmonic mean between recall and precision, results between 90.68% and 97.05%, and percentage of errors impostors accepted and legitimate users rejected, Equal Error Rate (EER), between 9.85% and 1.88% for static verification, login, and dynamics, post login, which point out the feasibility of using this proposed system as yet another layer of security, if used in conjunction with conventional methods such as a password.

Keywords: Continuous authentication, Behavioral biometrics, Mobile, Banking applications.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

level>

LISTA DE ACRÔNIMOS

AC	Acurácia
ACB	Acurácia Balanceada
ALG	Algoritmo
ALG(E)	Algoritmo e escopo
API	*Application Programming Interface*
CIAWI	Conferência Ibero-Americana WWW/Internet
CISTI	Conferência Ibérica de Sistemas e Tecnologias de Informação
Cc	Conta
Cc1	Conta Menu
Cc2	Conta Transação
C1	Cenário 1
C2	Cenário 2
C3	Cenário 3
EA	Escopo A
EB	Escopo B
EC	Escopo C
ED	Escopo D
EE	Escopo E
EF	Escopo F
EER	*Equal Error Rate*
F1	F1 *Score*
FAR	*False Acceptance Rate*
FAR_I	*FAR Impostores*
FN	*False Negative*
FP	*False Positive*
FRR	*False Rejection Rate*
GB	*Gradient Boosting*
GPS	*Global Positioning System*
GSMA	*Global System for Mobile Communication*
IGAE	*Information Gain Attribute Evaluator*
L	*Login*
KNN	*K-Nearest Neighbor*
MLP	*Multilayer perceptron*
MS	Menu Serviços
NB	*Naive Bayes*
NBB	*Naive Bayes Bernoulli*
NBG	*Naive Bayes Gaussian*

NN	*NeuralNet*
P	Pagamento
P1	Pagamento Menu
P2	Pagamento Transação
PRC	Precisão
PSO	*Particle Swarm Optimization*
QTD	Quantidade de *Templates*
REC	*Recall*
RF	*Random Forest*
SVM	*Support Vector Machine*
T	Transferência
T1	Transferência Menu
T2	Transferência Transação
TAR	*True Acceptance Rate*
TN	*True Negative*
TP	*True Positive*
TRR	*True Rejection Rate*
VE	Verificação Estática
VD	Verificação Dinâmica
XGB	*Extreme Gradient Boosting*

1 INTRODUÇÃO

De acordo com o GSMA *(Global System for Mobile Communication)*, órgão representante das operadoras de redes móveis a nível mundial, em seu relatório de 2019, no mundo já existem cerca de mais de 5 bilhões de pessoas que utilizam telefones *mobile* [1], deste total 3.2 bilhões eram *smartphones* [2] segundo dados da empresa Statista, empresa especializada em dados estatísticos. Segundo previsões da empresa de pesquisa Juniper, especializada em *mobile*, o número de usuários que utilizaram o *smartphone* para acessar aplicações *mobile* de bancos seria de 2 bilhões de pessoas em 2018 [3]. Com a ampla adoção destes dispositivos, o número de *malwares* específicos para dispositivos móveis também seguiu esta tendência. Segundo pesquisa do *Karspersky lab*, divulgada em 2019, o número de ataques a dispositivos móveis dobrou em 2018, superando os 116,5 milhões [4].

Esta migração natural para aplicações *mobile*, tem gerado uma evolução nos métodos de autenticação ao longo do tempo, visando garantir a prevenção de fraudes, principalmente no caso de aplicações críticas como as financeiras.

Comumente o primeiro ponto de interação com o dispositivo *mobile*, e com as aplicações é a autenticação. Existem três modos tradicionais usados para autenticação de uma pessoa: posse, conhecimento e biometria [5], algo que faz parte da pessoa. A autenticação por biometria acaba sendo o modo que implica em uma maior dificuldade ao fraudador, pois é necessário um esforço maior para conseguir forjar algo que faz parte da pessoa, em comparação com algo que a pessoa sabe ou possui. A autenticação por biometria pode ser dividida em fisiológica ou comportamental [6]: a fisiológica está ligada às características físicas das pessoas como, a íris e a digital; já as comportamentais estão ligadas às atitudes que são inerentes do indivíduo, por exemplo, a forma como caminha, ou como interage com um dispositivo *touchscreen*.

Neste trabalho foi proposto um modelo de autenticação contínua baseado em biometria comportamental para autenticação de usuários em aplicações bancárias *mobile*, com resultados de F1 *Score*, média harmônica entre o *recall* e a precisão, entre 90,68% e 97,05%, e percentual de erros referentes a impostores aceitos e usuários legítimos rejeitados, *Equal Error Rate* (EER), entre 9,85% e 1,88% para verificação estática, *login*, e dinâmica, pós *login*, que apontam a viabilidade do uso deste sistema proposto como mais uma camada de segurança, se utilizado em conjunto com métodos convencionais como senha.

1.1 DEFINIÇÃO DO PROBLEMA

Definir uma autenticação baseada em senha, é o método mais frequentemente utilizado para proteger dados de intrusos [7]. Mas usualmente, só é solicitado ao usuário a autenticação via

senha, ou outra credencial, no primeiro momento de interação com a aplicação, no *login*. Uma possibilidade para aumentar a segurança do processo de autenticação é autenticar o usuário durante todo o momento de interação com uma aplicação e não apenas no *login*. Neste contexto, as abordagens de autenticação contínua podem fornecer uma linha adicional de defesa, concebidas como uma contramedida de segurança não intrusiva e passiva [8], pois esta pode acontecer de forma implícita, gerando um maior equilíbrio entre segurança e usabilidade.

Com base neste contexto, este trabalho propõe um *framework* multimodal de autenticação contínua e implícita, baseado no padrão de comportamento biométrico na interação com o *touchscreen*, na localização capturada via GPS, e nas informações coletadas de vários sensores, para aplicações bancárias *mobile*, visando a detecção de possíveis fraudadores, e geração de alertas caso isso aconteça. A expectativa é que a união destes fatores de autenticação biométrica com o padrão de localização possa cobrir todo o momento de interação do indivíduo com a aplicação, garantindo um modelo de autenticação contínua realmente eficaz e de alto desempenho, com melhores resultados dos que o já observados anteriormente na literatura, que variaram entre 82,53% e 96% de acurácia e 0 e 11,5% de EER.

Os esquemas de autenticação implícita utilizam biometria comportamental para autenticar de forma contínua e transparente [9]. Por exemplo, a biometria comportamental pode ser capturada através da interação do usuário com o *touchscreen* do celular. Do toque do dedo humano em contato com a tela podem ser capturadas características que identificam o indivíduo de forma única [10]. Estudos recentes revelam que os sensores dos *smartphones* tem um rico potencial para serem utilizados na autenticação ativa/contínua [11].

1.2 OBJETIVOS

Nesta seção serão detalhados o objetivo geral e os específicos que nortearão o desenvolvimento deste trabalho.

1.2.1 Objetivo geral

Desenvolver um *framework* de autenticação contínua para aplicações bancárias *mobile*, baseado no padrão de comportamento biométrico na interação com o *touchscreen*, na localização capturada via GPS, e nas informações coletadas de vários sensores, para detecção de anomalias e geração de alertas.

1.2.2 Objetivos específicos

- Identificar qual a quantidade mínima de interações necessárias para treino do modelo, tornando possível autenticar um usuário ao digitar uma senha.

- Encontrar qual a quantidade mínima de interações necessárias para treino do modelo, tornando possível autenticar um usuário a partir da interação com uma aplicação após *login*.
- Validar se o padrão de digitação e interação do usuário com uma aplicação *mobile* é mantido de forma consistente entre sessões.
- Detectar se as informações capturadas de sensores são realmente relevantes para a definição de um modelo de autenticação contínua baseada em biometria comportamental.
- Evidenciar quais as características mais discriminantes, quando são incluídos dados de vários sensores para autenticação contínua baseada em biometria comportamental.
- Constatar qual algoritmo terá a melhor desempenho para a criação de um modelo de Aprendizado de Máquina baseado em biometria comportamental com base no F1 Score.
- Investigar qual a quantidade de interações necessárias com uma aplicação para que seja possível autenticar um usuário com um F1 *score* a partir de 90%.

1.3 PRINCIPAIS CONTRIBUIÇÕES

Esse trabalho propõe um *framework* multimodal baseado em biometria comportamental e localização para autenticação continua de usuários em aplicações bancárias móveis. O *framework* une os padrões de digitação e de deslize de tela com a localização GPS do usuário no processo de autenticação, utilizando pelo menos 6 algoritmos de Aprendizado de Máquina diferentes. Durante o desenvolvimento da proposta do *framework* foram gerados duas publicações, sendo elas:

Artigo curto no CIAWI:

- **Referência:** P. M. A. B. Estrela, D. M. Amaral, R. de Oliveira Albuquerque, W. F. Giozza, G. D. A. Nze and F. L. L. de Mendonça, "Estudo experimental da bometria comportamental para autenticação contínua de usuários em aplicações bancárias *mobile*,"2019 16ª conferência Ibero-Americana WWW/Internet (CIAWI), Lisboa, Portugal, 2019, pp. 248-252, ISBN: 978-989-8533-96-8;
- **Principais contribuições**: análise de desempenho algoritmos *K-Nearest Neighbors* e *Random Forest*, com melhor desempenho observado para *Random Forest*.
- **link:** http://www.iadisportal.org/digital-library/estudo-experimental-da-biometria-comportamental-para-autentica%C3%A7%C3%A3o-cont%C3%ADnua-de-usu%C3%A1rios-em-aplica%C3%A7%C3%B5es-banc%C3%A1rias-mobile.

Artigo completo no CISTI 20:

- **Referência:** P. M. A. B. Estrela, R. de Oliveira Albuquerque, D. M. Amaral, W. F. Giozza, G. D. A. Nze and F. L. L. de Mendonça, "Biotouch: a framework based on behavioral

biometrics and location for continuous authentication on mobile banking applications,"2020 15th Iberian Conference on Information Systems and Technologies (CISTI), Sevilla, Spain, 2020, pp. 1-6, doi: 10.23919/CISTI49556.2020.9140948;

- **Principais contribuições**: inclusão de coleta de dados a partir de mais sensores, análise de resultado com novos algoritmos de Aprendizado de Máquina, melhor resultado observado com *Naive Bayes Bernoulli* e *Naive Bayes Gaussian* e análise das características mais importantes;
- **link:** https://ieeexplore.ieee.org/document/9140948.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está estruturado como se segue. No Capítulo 2 são apresentados os principais conceitos que norteiam o desenvolvimento do *framework* proposto e uma breve revisão da literatura, listando os trabalhos correlatos e justificando a proposta deste trabalho. No Capítulo 3 é apresentado o modelo proposto, incluindo uma descrição detalhada da estrutura e do funcionamento do *framework* proposto. No Capitulo 4 é discutida uma proposta de implementação do *framework* e de experimentos. No Capítulo 5 são apresentadas as conclusões, incluindo sugestões para possíveis trabalhos futuros.

2 CONCEITOS E REVISÃO DA LITERATURA

Neste capítulo será feita uma revisão da literatura e trabalhos correlatos que fundamentam o modelo de *framework* proposto.

2.1 BIOMETRIA COMPORTAMENTAL *TOUCH*

A biometria comportamental *touch* se refere ao processo de medir e avaliar o ritmo do toque humano em dispositivos *touchscreen mobile* [12]. Para definir o comportamento biométrico quando se utiliza um dispositivo *touchscreen*, as informações coletadas dos vários sensores que compõem os *smartphones* modernos como acelerômetro, sensor de luz do ambiente, compasso de digitação, giroscópio, GPS, sensor de proximidade, *touchscreen* e WiFi [13], são capturadas para construírem um padrão biométrico do indivíduo.

No contexto das aplicações *mobile*, a biometria comportamental surge como um modelo de biometria menos intrusivo e que pode ser capturado de forma implícita, provendo um maior equilíbrio entre segurança e usabilidade, especialmente a *touch*, que não faz parte das informações privadas do usuário.

2.2 CARACTERÍSTICAS DA BIOMETRIA *TOUCH*

As características biométricas geradas via interação *touchscreen*, além de serem menos intrusivas, contêm atributos úteis, conforme listado na Tabela 2.1, se comparado a outros métodos biométricos [12].

Tabela 2.1: Atributos das características da biometria *touch* [12]

Tipo	Descrição
Distintividade	A dinâmica do toque em uma tela *touchscreen* pode gerar eventos que são capturados pelos sensores do aparelho, oferecendo informações de tempo, espaço e movimento que identificam o indivíduo unicamente e que são difíceis de serem replicadas.
Aumentar a segurança	Integrar a biometria *touch* com a senha pode elevar a segurança do sistema.
Monitoração contínua	O comportamento do usuário pode ser monitorado durante todo o tempo de interação com a aplicação para reautenticação de uma forma fácil e não intrusiva.

Atributo	Justificativa
Revogabilidade	Quando um padrão biométrico salvo em conjunto com uma senha for comprometido, um novo *template* pode ser facilmente gerado quando uma nova senha for cadastrada.
Não dependência	O processo de aquisição de características *touch* é pouco sensível aos fatores do ambiente, diferentemente de biometria de face ou voz.
Transparência	A aquisição das características da dinâmica *touch* pode acontecer em segundo plano, enquanto o usuário interage de forma natural com a aplicação no dispositivo *mobile*.
Familiaridade	Os dados colhidos sobre a dinâmica *touch* do indivíduo podem ser capturados de forma simples e natural durante a interação com a aplicação.
Relação custo benefício	Não é necessária a utilização de nenhum sensor especial além dos que já estão incorporados no dispositivo, diferente de biometrias como íris e digital.

Mas a coleta dessas características também impõe desafios, devido às peculiaridades e limitações dos dispositivos *mobile*. Entre estes desafios podemos citar: minimizar o custo computacional e da comunicação; minimizar o consumo de energia; maximizar a acurácia do modelo; e, preocupar-se com a capacidade de adaptação [12]. Também deve-se levar em conta que a biometria comportamental *touch* pode sofrer mudanças devido a fatores externos, como mudanças emocionais, estado físico e métodos de aquisição pobre [14], por exemplo, um dispositivo com sensores de baixa qualidade que tornaria difícil a captura de dados precisos.

2.2.1 Tipos de extração das características de biometria *touch*

A extração de características de biometria *touch* (capturas) pode ser feita de diferentes maneiras: espacial, do movimento [12], temporal, dinâmico e geométrico [15], conforme listadas na Tabela 2.2.

Tabela 2.2: Tipos das características da biometria *touch*.

Tipo	Descrição
Espacial	Extração associada com as interações físicas entre as pontas dos dedos e a superfície de um dispositivo *touchscreen*. As três características mais comuns são: tamanho do *touch*, espaço que o dedo ocupa na tela; pressão e posição, coordenadas x e y do local onde o dedo toca a tela [12].
Temporal	Extraídas da análise do tempo em que os movimentos acontecem: o tempo total de um evento *touch* pode ser calculado pela diferença entre o tempo inicial e o tempo final [15].

Tipo	Descrição
Dinâmico	Extraídas da análise dinâmica do movimento *touch*, por exemplo, um evento *touch* é gerado a partir do toque do dedo na tela [15]. Este movimento deve ser analisado e interpretado para que sejam geradas características capazes de diferenciar os indivíduos.
Geométrico	Extraídas via uma análise geométrica de um evento *touch*. Um evento *touch* contém um ou mais toques, e o toque é uma sequência de pontos e tempos. A análise da relação entre estes pontos, linhas e curvas, pode gerar características discriminativas [15].

2.2.2 Implementação da captura das características da biometria *touch*

Para a captura das características de biometria *touch* em dispositivos *mobile*, são utilizadas informações disponibilizadas pelas APIs dos sistemas operacionais, que são executados nos dispositivos. Por exemplo, para os dispositivos Android, foco desse trabalho, essas informações podem ser capturadas por meio das APIs *MotionEvent e SensorEvent*.

A Tabela 2.3 detalha os métodos e atributos geralmente utilizados para a *MotionEvent* [16] e a Tabela 2.4 para a *SensorEvent* [17], respectivamente.

Tabela 2.3: Métodos MotionEvent para captura de características *touch*.

Característica	Metódo	Sensor
Tempo	*getEventTime(), getDownTime()*	*Touchscreen*
Ação	*getAction()*	*Touchscreen*
Pressão	*getPressure()*	*Touchscreen*
Tamanho do dedo	*getSize()*	*Touchscreen*
Coordenada x	*getX()*	*Touchscreen*
Coordenada y	*getY()*	*Touchscreen*

Tabela 2.4: Métodos SensorEvent para captura de características *touch*

Característica	Sensor	Atributo
Força de aceleração ao longo do eixo X (incluindo a gravidade) [18].	Acelerômetro	SensorEvent. values[0]
Força de aceleração ao longo do eixo Y (incluindo a gravidade) [18].	Acelerômetro	SensorEvent. values[1]
Pressão Força de aceleração ao longo do eixo Z (incluindo a gravidade) [18].	Acelerômetro	SensorEvent. values[2]

Característica	Sensor	Atributo
Taxa de rotação ao redor do eixo X [18].	Giroscópio	SensorEvent. values[0]
Taxa de rotação ao redor do eixo Y [18].	Giroscópio	SensorEvent. values[1]
Taxa de rotação ao redor do eixo Z [18].	Giroscópio	SensorEvent. values[2]
Campo geomagnético do ambiente para o eixo físico X em T [19].	Magnetômetro	SensorEvent. values[1]
Campo geomagnético do ambiente para o eixo físico Y em T (x, y, z) em T [19].	Magnetômetro	SensorEvent. values[1]
Campo geomagnético do ambiente para o eixo físico Z em T (x, y, z) em T [19].	Magnetômetro	SensorEvent. values[2]
Componente do vetor de rotação ao longo do eixo X (X* sen (/ 2)) [18].	Sensores de rotação (software ou hardware)	SensorEvent. values[0]
Componente do vetor de rotação ao longo do eixo Y (Y* sen (/ 2)) [18].	Sensores de rotação (software ou hardware)	SensorEvent. values[1]
Componente do vetor de rotação ao longo do eixo Z (Z* sen (/ 2)) [18].	Sensores de rotação (software ou hardware)	SensorEvent. values[2]
Componente escalar do vetor de rotação ((cos (/ 2)) [18].	Sensores de rotação (software ou hardware)	SensorEvent. values[3]
Acurácia estimada	Sensores de rotação (software ou hardware)	SensorEvent. values[4]
Força de aceleração ao longo do eixo X (excluindo a gravidade) [18].	Sensores de aceleração (software ou hardware)	SensorEvent. values[0]
Força de aceleração ao longo do eixo Y (excluindo a gravidade) [18].	Sensores de aceleração (software ou hardware)	SensorEvent. values[1]
Força de aceleração ao longo do eixo Z (excluindo a gravidade) [18].	Sensores de aceleração (software ou hardware)	SensorEvent. values[2]
Força da gravidade ao longo do eixo X [18].	Sensores de gravidade (software ou hardware)	SensorEvent. values[0]
Força da gravidade ao longo do eixo Y [18].	Sensores de gravidade (software ou hardware)	SensorEvent. values[1]

Característica	Sensor	Atributo
Força da gravidade ao longo do eixo Z [18].	Sensores de gravidade (software ou hardware)	SensorEvent. values[2]

A partir dos dados capturados via APIs Android, outras características podem ser derivadas, para formarem os vetores de informações que irão ser armazenados em forma de *templates* de identificação do indivíduo, como por exemplo: valores do total de tempo do movimento; velocidade média; mediana; valores máximos e mínimos para a pressão aplicada durante o movimento; entre outros. A qualidade dos dados (capturados e derivados) são de fundamental importância para a boa performance do sistema de autenticação contínua.

2.3 AUTENTICAÇÃO CONTÍNUA

A autenticação é um meio de garantir a verificação da identidade de um indivíduo durante a interação com um dispositivo computacional. É utilizada para evitar que um impostor, não autorizado [20], obtenha acesso como um usuário legítimo.

Usuários podem ser autenticados via uma das seguintes políticas [21]:

- **Baseada em conhecimento**: algo conhecido, como uma senha.
- **Token ou Posse**: algo que está sob a propriedade do usuário, por exemplo, *smart-card*;
- **Biometria**: algo que faz parte da pessoa, nesse caso o usuário fornece seus atributos físicos ou comportamentais para que seja autenticado.

O método de autenticação mais utilizado é o de conhecimento, normalmente senha. E geralmente a senha só é solicitada no primeiro momento de interação com a aplicação, caracterizando uma autenticação estática. Mas a autenticação também pode ser contínua/ativa, ou seja, acontecer durante todo o momento de interação do indivíduo com um dispositivo computacional, no caso deste trabalho, o *smartphone*.

A autenticação ativa pode ser definida como a contínua verificação da identidade de uma pessoa baseado em aspectos do seu comportamento na interação com um dispositivo computacional [22].

As principais características da autenticação ativa são [15]:

- **Continuidade**: o usuário é autenticado durante todo o momento em que interage com o dispositivo, através de processos de reautenticação que acontecem de forma periódica;
- **Transparência**: todo o processo de autenticação pode ser feito em segundo plano sem interromper o usuário em suas atividades.

2.3.1 Métodos de operação da autenticação contínua

O processo de autenticação pode ter duas formas de operação, via identificação ou via verificação [12], [15]. Na identificação, o sistema reconhece o *template* biométrico apresentado, comparando com todos os que estão armazenados [15], o reconhecimento é feito "1-para-N". No caso da identificação a pergunta a ser respondida é: "quem é esta pessoa?" ou "esta pessoa está na base?" [12]. Já na verificação, o usuário apresenta a identidade, e o processo verifica se este é quem afirma ser, o reconhecimento é feito "1-para-1", e a pergunta a ser respondida é: "esta pessoa é quem ela afirma ser?" [12]. Na operação via verificação, a autenticação pode acontecer em dois modos, estático ou dinâmico. No modo estático, a autenticação biométrica do usuário acontece no momento de *login*, ou em um intervalo de tempo definido durante a sessão. No modo dinâmico, as características capturadas refletem as informações geradas pelo usuário em tempo real durante a sessão, mas o intervalo entre as várias autenticações não são pré-definidos, eles podem ser determinados pela ocorrência de alguns eventos de *touch* [12]. Sendo assim os intervalos para as várias verificações dinâmicas, durante o período em que o usuário estiver com a sessão aberta, podem ser definidos, com base em um período de tempo ou em uma quantidade, pré definida, de movimentos.

Na autenticação contínua em dispositivos *mobile* o processo de verificação pode acontecer baseado em um comportamento único, como o padrão de digitação ou pode se dar de forma multimodal, observando um conjunto de vários comportamentos e classificadores para a definição da biometria comportamental do indivíduo [22]. Para a identificação do padrão de comportamento de um indivíduo de forma contínua, é necessário que sejam definidas quais características serão capturadas, estas posteriormente servirão de entrada para um algoritmo de inteligência artificial que será capaz de determinar a probabilidade de identificação de um indivíduo, baseando-se nas características que estão sendo colhidas durante o momento de interação com a aplicação.

2.3.2 Composição de um *framework* para autenticação contínua

Um *framework* de autenticação contínua é composto de três fases:

(i) Registro do usuário [12]: Fase na qual é realizado o cadastro do usuário no sistema;

(ii) Autenticação do usuário [12]: Fase em que é realizada a verificação do usuário, baseada nos modelos salvos no sistema para este fim;

(iii) Atualização dos dados [12]: Fase de adaptação do *template* registrado para um usuário.

Na Tabela 2.5 são detalhados os módulos que compõem o sistema, durante as três fases do processo de autenticação ativa.

Tabela 2.5: Detalhamento dos módulos que compõem um sistema de autenticação contínua.

Módulo	Descrição	Fase
Aquisição dos dados [12] [15]	Captura das informações dos usuários.	i, ii
Pré-processamento dos dados [12]	Transformações necessárias sobre os dados (ex: redução de dimensões, normalizações e remocão de *outliers*).	i, ii
Extração das características [12] [15]	Extração das características desejadas com base nos dados capturados nas fases anteriores.	i, ii
Geração dos *templates* [12]	Processo de transformação das características capturadas em um *template* de identificação do usuário que será armazenado. Este *template* pode ser formado por vários vetores de informações que identificam o usuário.	i
Classificação dos dados [12]	Informações são categorizadas e comparadas com os *templates* de referência, utilizando algoritmos de aprendizado de máquina [12].	i
Tomada de decisão [12]	Onde é realizada a decisão, com base no resultado obtido na fase anterior, sobre a legitimidade da identificação requerida.	ii
Adaptação dos dados [12]	A base de *templates* precisa sofrer adaptações, pois as características do comportamento biométrico do indivíduo modificam ao longo do tempo. É necessário estabelecer uma política para a atualização dos *templates* dos usuários.	iii

As principais técnicas de Aprendizado de Máquina reportadas para dinâmicas *touch* são: Modelagem probabilística, Análises de *clusters*, Árvores de decisão, *Support Vector Machine* (SVM) e Redes neurais [12].

2.3.3 Aprendizado de máquina e autenticação contínua

Para a criação dos modelos que serão utilizados para a autenticação contínua de usuários são utilizados algoritmos de Aprendizado de Máquina. O Aprendizado de Máquina é uma ciência emergente para construir programas capazes de imitar o comportamento humano e tomar decisões eficientes como um humano [23]. Sendo definida também como a criação e uso de modelos que são aprendidos a partir de dados [24], tornando possível que esse aprendizado seja adaptado para diferentes cenários [25].

Esse aprendizado pode ser dividido em supervisionado e não supervisionado [23] [24] [25], o aprendizado supervisionado pode ser definido como aquele nos quais existe um conjunto de dados

etiquetados com a resposta correta para aprendizagem [24], isso significa que cada exemplo que é utilizado para alimentar o algoritmo está classificado em uma reconhecível classe de dados [25]. O não supervisionado é aquele no quais não existem as etiquetas[24], neste tipo de aprendizado é fornecido milhares de problemas e os seus resultados, mas não é explicado como o resultado é calculado. O algoritmo então inicia a procura por coisas em comum, e tenta identificar traços entre os problemas e as soluções. Com a quantidade de dados suficientes o algoritmo vai identificar padrões e desenvolver uma estratégia para solucionar o problema [25].

Um perigo comum em aprendizado de máquina é o sobreajuste, *overfitting*, e o sub-ajuste, *underfitting*. O *overfitting* acontece quando é produzido um modelo de bom desempenho com os dados de treino, mas que não lida muito bem com os novos dados. O *underfitting* acontece quando é produzido um modelo que não desempenha bem nem com os dados de treino [24].

As principais vantagens de utilização de aprendizado de máquina são [23]:

- Fácil identificação de tendências e padrões ocultos em grandes conjuntos de dados que seriam tediosos para os seres humanos detectarem;
- Não é necessária intervenção ou conhecimento humano para aprender com os dados;
- É capaz de lidar com conjuntos de dados multi-variáveis e multidimensionais.

O aprendizado de máquina também tem as suas desvantagens, entre elas se pode citar [23]:

- É necessário adquirir um conjunto de dados grande com boa qualidade e imparcial;
- Existe o desafio de interpretar os resultados e encontrar os erros, pois sem a interpretação os dados não são úteis;

O Aprendizado de Máquina supervisionado pode ser dividido em classificação e regressão [23] [25], a classificação é feita principalmente quando os dados de saída são uma variável discreta, por exemplo, homem ou mulher, fuma não fuma [23], usuário legítimo ou impostor. E na regressão o resultado é uma variável contínua, por exemplo, prever o peso do próximo ovo produzido em uma fazenda de criação de aves [25].

Entre os mais relevantes algoritmos de *Machine Learning* [25], entre os que podem ser utilizados para classificação podemos citar:

- *K-Nearest Neighbor* (KNN): é um dos mais simples algoritmos de aprendizado de máquina, sempre que faz predições ele utiliza todo o *dataset* [25]. A previsão para cada ponto novo depende somente de alguns pontos mais próximos [24].
- *Naive Bayes* (NB): é baseado em modelagem estatística para lidar com problemas de classificação [25];

- Árvores de Decisão: são utilizadas para criar modelos de classificação baseada em estrutura de árvores. É um algoritmo extremamente fácil para leitura e entendimento [25]. Mas as árvores de decisão tem uma tendência a *overfitting* que pode ser corrigida com a utilização de outros algoritmos baseados em árvores, como o *Random Forest* (RF), que com a utilização de *bagging* soluciona essa questão [24] e o *Gradient Boosting* (GB), que utilizando da técnica de *boosting* tenta resolver a questão da tendência das árvores de decisão. No *bagging* são utilizados múltiplos modelos, gerados de subconjuntos diferentes dos dados e no *boosting* cada modelo criado aprende a fazer a correção dos erros gerados pelo modelo anterior;
- *Support Vector Machine* (SVM): ele busca pela melhor linha que separa todas as diferentes classes presentes nos dados de treino [25].
- Redes Neurais: é um dos grupos mais poderosos de Aprendizado de Máquina, mas também são mais complexos. É derivado da estrutura e processos do nosso próprio cérebro, dos neurônios [25]. Consistem em neurônios artificiais, que desenvolvem cálculos similares sobre suas entradas [24]. Para a maioria dos problemas, elas provavelmente não são a melhor opção [24], pela sua complexidade. Elas são muito usadas em *Deep Learning*, Aprendizado Profundo [24].

Sobre a utilização de algoritmos de aprendizagem de máquina em autenticação contínua baseada em biometria comportamental *touch*, na literatura é mencionado a utilização de várias técnicas diferentes, sendo [12]:

- Modelagem probalística: *Bayesian Network*, *Naive Bayes* e *Gaussian Probability Density Function*;
- *Cluster*: *K-means*, *K-Star* e *k-Nearest Neighbors* (KNN);
- Árvores de decisão: *J48* e *Random Forest* (RF);
- *Support Vector Machine* (SVM);
- Redes neurais; *Radial Basis Function Networks*
- Medidas de distância: Euclidiana, *Mahattan*, *Mahalanobis* e *Bhattacharry*.
- Estatístico;

2.3.4 Métricas de sistemas de autenticação contínua

Os sistemas de autenticação contínua baseados em biometria possuem métricas para a mensuração da sua eficiência, tais como:

- *True Positive* (TP): se refere a correta predição de uma classe legítima:
  classe 1 –> predição 1

- *False Positive* (FP): se refere a classificação incorreta de uma classe ilegitima como legítima:
classe 0 –> predição 1

- *False Negative* (FN): se refere a classificação incorreta de uma classe legítima como ilegítima:
classe 1 –> predição 0

- *True Negative* (TN): se refere a classificação correta de uma classe ilegítima:
classe 0 –> predição 0

- *False Acceptance Rate* (FAR): se refere à porcentagem de impostores que foram aceitos pelo sistema [26], [27], [28], [29]. Essa taxa pode ser encontrada calculando a razão entre o de número de *templates* de impostores que foram aceitos, dividido pelo número total de *templates* dos impostores [27]:

$$\text{FAR} = \frac{\text{FP}}{\text{FP+TN}}\text{x}100 \tag{2.1}$$

- *False Rejection Rate* (FRR): se refere à porcentagem de usuários autênticos que foram rejeitados pelo sistema [26], [27], [28], [29]. Esse valor pode ser encontrado dividindo o número de *templates* de usuários legítimos que foram rejeitados, pelo número total de *templates* dos usuários legítimos [27]:

$$\text{FRR} = \frac{\text{FN}}{\text{FN+TP}}\text{x}100 \tag{2.2}$$

- *Equal Error Rate* (EER): se refere ao ponto em que os erros referentes a impostores aceitos e autênticos rejeitados, são iguais [26], [27], [28], [29]. O EER torna fácil comparar a desempenho de vários sistemas biométricos ou classificadores, quanto menor o valor melhor é o classificador [30]. O cálculo desse valor, utilizando-se o FAR e o FRR com a menor diferença, pode ser feito [29]:

$$\text{EER} = \frac{\text{FAR+FRR}}{2} \tag{2.3}$$

- *True Acceptance Rate* (TAR): se a refere à porcentagem de usuários que foram corretamente aceitos pelo sistema [27]. Esse valor pode ser encontrado pela equação [31]:

$$\text{1-FRR} \tag{2.4}$$

- *True Rejection Rate* (TRR): se refere à porcentagem de usuários que foram corretamente rejeitados pelo sistema [27];

- *Recall* (REC): é conhecida também como sensibilidade, ou revocação, e se refere à proporção de casos positivos que foram corretamente previstos [32]. Esse valor pode ser encontrado pela equação [24]:

$$\text{REC} = \frac{\text{TP}}{\text{TP+FN}}\text{x}100 \tag{2.5}$$

- Precisão (PRC): denota a proporção de casos positivos previstos que realmente são casos positivos [32]. Esse valor pode ser encontrado pela equação [24]:

$$\text{PRC} = \frac{\text{TP}}{\text{TP+FP}}\text{x}100 \tag{2.6}$$

- F1 *Score* (F1): é definido como a média harmônica do *Recall* e da precisão [33]. O modelo é tanto melhor quanto maior o valor de F1 *Score* [34]. Esse valor pode ser encontrado pela equação [24]:

$$\text{F1} = \frac{2\text{x}(\text{PRCxREC})}{(\text{PRC+REC})} \tag{2.7}$$

- Acurácia (AC): se refere à taxa de acerto geral do sistema [27]. Esse valor pode ser encontrado, calculando a razão entre o número de *templates* de usuários legítimos que foram aceitos e impostores que foram rejeitados, pelo número total de templates dos usuários [27] [24]:

$$\text{AC} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}\text{x}100 \tag{2.8}$$

- Acurácia Balanceada (ACB): Pode ser definido como a precisão média obtida em qualquer classe [35], é uma média simples entre *Precision* e *Recall*. Para o cálculo do valor a seguinte equação pode ser utilizada:

$$\text{ACB} = \frac{\text{PRC+REC}}{2}\text{x}100 \tag{2.9}$$

2.4 TRABALHOS CORRELATOS

Nesta seção será feita a revisão da literatura dos trabalhos que serviram de fundamento para o desenvolvimento do modelo de *framework* proposto nessa dissertação.

2.4.1 Verificação estática

M. Antal, L. Z. Szabo, e I. Laszlo, em [7], apresentam uma aplicação para Android com a finalidade de coletar dados do usuário na fase de registro. Levando em consideração que o padrão de digitação pode ser influenciado por diversos fatores, os dados foram coletados em mais de uma sessão. Os aparelhos utilizados no experimento foram um Nexus 7 tablet e Mobil LG Optimus L7 II.

Os usuários precisavam digitar a mesma senha considerada forte, por 30 vezes em cada sessão. A senha escolhida (.tie5Roanl) exigia o pressionamento de 14 teclas, sendo 8 letras, um dígito, um ponto, duas vezes a tecla *Shift*, a fim de alternar para letras maiúsculas e duas vezes a tecla de teclado numérico. Participaram desse estudo 42 pessoas, todas usaram a mesma senha para que os dados de cada participante pudessem ser usados como um impostor e como um usuário legítimo na hora do treinamento.

Como classificadores foram selecionados alguns algoritmos implementados com o Weka, *software* para trabalhar com Aprendizado de Máquina, cobrindo 7 métodos de Aprendizado de Máquina diferentes: *Naive Bayes, Bayesian Network, Nearest neighbors* (KNN, IBk no Weka), J48 (C4.5 no Weka), SVM, *Random Forest* e *Multilayer perceptron* (MLP). Os sete métodos foram aplicados utilizando 41 e 71 características, geradas a partir da dinâmica de digitação da informação alvo. A Tabela 2.6 apresenta os componentes do vetor de características base utilizado para o caso das 71 características. O algoritmo que apresentou melhor performance foi o *Random Forest*, tanto com 41 e 71 características, 82,53% e 93,04% de acurácia, respectivamente.

Tabela 2.6: Componentes do vetor de características capturadas em [6]

Característica	**Descrição**
Key hold time	Tempo entre pressionar e soltar a tecla
Down-down time	Tempo entre pressionamentos de tecla consecutivos
Up-down time	Tempo entre a soltura de uma tecla e o pressionamento da próxima
Key hold pressure	Pressão no momento de pressionamento da tecla
Finger area	Área do dedo no momento de pressionamento da tecla
Average hold time	Média do tempo entre pressionamento e soltura das teclas
Average finger area	Média dos valores de área do dedo ao pressionar as teclas
Average pressure	Média da pressão ao pressionar as teclas

Através do conjunto de dados coletados relativos à interação dos usuários com *touchscreen*, utilizando características baseadas em informações espaciais (tamanho, pressão, tempo e posição [12]), esse trabalho [7], demonstrou experimentalmente que as características colhidas a partir da digitação dos usuários em dispositivos *touchscreen*, podem ser utilizadas para a autenticação em aplicações *mobile* de uma forma menos intrusiva e efetiva visto que obteve uma acurácia de até 93,04%.

2.4.2 Verificação Dinâmica

Em [8], M. Frank et al. propõem uma experiência para captura de gestos dos usuários, em que cada convidado deveria interagir com uma aplicação, onde deveria ler 3 documentos sobre assuntos diversos e interagir com 2 imagens para encontrar diferenças. O propósito era colher dos usuários dados da interação com *touchscreen* referentes aos movimentos de deslizamento horizontal e o deslizamento vertical, durante a interação com o conteúdo em deslizamentos para cima ou para baixo, o que pode acontecer durante a navegação com páginas web, ou aplicações em

geral. Participaram desta pesquisa 41 voluntários, e os aparelhos utilizados foram 5 *smartphones* diferentes: Droid Incr., Nexus One, Nexus S, Galaxy S, todos com Android 2.3.x.

Neste experimento de cada *stroke*, uma sequência que começa com o toque na tela e termina quando o dedo é levantando, foram originadas 30 características diferentes, derivadas das informações capturadas via API Android listadas na Tabela 2.7.

Tabela 2.7: Características capturadas via API Android em [8]

Característica
Tempo
Ação
Orientação do telefone
Coordenada x
Coordenada y
Pressão
Área do dedo
Orientação do dedo

Os classificadores utilizados foram o SVM e o KNN. O experimento foi proposto em três momentos distintos: entre semanas, entre sessões e autenticação de curto prazo. Teve uma média de EER entre 0%, para intra sessões, 2%-3% para inter sessões e 4% quando a autenticação aconteceu uma semana depois do registro. No experimento o SVM apresentou uma taxa de erro menor que o KNN.

Com base nos resultados do experimento [8] ficou comprovado que é possível distinguir usuários baseando-se no modo como estes interagem com o *touchscreen*, com base no deslocamento horizontal e vertical do dedo na tela.

2.4.3 Verificação estática em aplicações bancárias *mobile*

A. Buriro, S. Gupta e B. Crispo, em [36], descrevem um experimento com autenticação contínua em aplicações *mobile* com foco em Internet *Banking*. Para a captura das características foi desenvolvida uma aplicação, bem parecida com uma aplicação original de uma solução de meio de pagamento bem conhecida, que funciona em qualquer dispositivo com Android 4.4.x ou maior. Os voluntários foram recrutados via plataforma "UBERTESTERS". Cada voluntário precisou digitar a senha de 8 dígitos, em 3 sessões durante 3 dias. Foram coletados dos 95 usuários, 30 amostras de cada. Foram coletadas informações do acelerômetro, de orientação, sensor de gravidade, magnetômetro e giroscópio. Com base nestas informações foram geradas 142 características diferentes, para cada indivíduo. Para a seleção das características mais relevantes foi utilizado o algoritmo de *Information Gain Attribute Evaluator* (IGAE).

Foram utilizados os classificadores *Naive Bayes* (NB), *NeuralNet* (NN), e RF. O melhor resultado observado durante os experimentos foi uma acurácia de 96% utilizando o algoritmo *Random*

Forest e 15 amostras para treino. Neste trabalho [36] não foram definidos aparelhos específicos para o experimento, apenas a versão do sistema operacional Android suportada foi o requisito para instalar a aplicação desenvolvida e participar do experimento, demonstrando a possibilidade da não definição do tipo de dispositivo em um experimento de autenticação contínua baseada em biometria comportamental *touch*, e obtendo, mesmo assim, uma alta acurácia.

2.4.4 Verificação dinâmica e estática em aplicações bancárias *mobile*

Em [37], M. Temper, S. Tjoa e M. Kaiser, apresentam um *framework* visando garantir segurança para aplicações bancárias *mobile*, a partir da autenticação contínua de usuários. O trabalho envolveu 22 voluntários, que interagiram com uma aplicação prototipada, baseada na interface de uma aplicação original de um banco. Foram colhidas 30 sessões de cada usuário, 10 foram utilizadas para treino e as outras 20 para testes. O aparelho utilizado foi um Nexus 4 com Android 4.4.4.

A identificação do usuário foi baseada na dinâmica de digitação e na interação com o *touchscreen*. Para o experimento em [37] foram utilizadas 15 características diferentes conforme a Tabela 2.8, selecionadas as mais relevantes via *Particle Swarm Optimization* (PSO).

Tabela 2.8: Características coletadas em [37]

Característica	**Descrição da característica**
Tamanho	A média do tamanho do dedo durante um toque
Tamanho mínimo	O tamanho mínimo ocupado pelo dedo na tela durante um toque
Tamanho máximo	O tamanho máximo ocupado pelo dedo na tela durante um toque
Diferença de tempo	Diferença de tempo entre o início e o fim de um toque
Ângulo	O ângulo entre o início e o fim das coordenadas do movimento do toque
X inicial	A coordenada x onde o movimento começou
Y inicial	A coordenada y onde o movimento começou
X final	A coordenada x onde o movimento terminou
Y final	A coordenada y onde o movimento terminou
Distância	A distância entre o início e o fim do toque
Ângulo de postura x	Ângulo de inclinação do telefone em relação a x
Ângulo de postura z	Ângulo de inclinação do telefone em relação a z

A principal contribuição deste trabalho [37] foi utilizar um classificador baseado na lógica de *Fuzzy*, resultando num EER de 11,5%, um valor alto em relação aos trabalhos anteriormente apresentados. Mas apesar disto, este trabalho correlato foi considerado por ter o foco em aplicações bancárias, e sua estrutura conter a captura tanto estática quanto dinâmica da biometria comportamental do indivíduo, servindo como um parâmetro de referência, quanto à busca por uma melhor acurácia.

2.4.5 Localização

A inclusão de informações do contexto do usuário no processo de autenticação contínua pode contribuir como um fator a mais para verificação do padrão de utilização de uma aplicação *mobile* por um usuário. Entre as informações de contexto temos a localização. Em [38] e [22], a localização é um dos fatores que compõem os esquemas de autenticação multimodal para aplicações *mobile*. Em [38] os dados de localização dos usuários foram utilizados em conjunto com as informações de movimento do usuário e de uso do aparelho. Para cada usuário foram estabelecidos dois perfis, um para os dias da semana e outro para os fins de semana. O padrão foi definido com base no histórico de cada usuário. Para o agrupamento dos dados da localização foi utilizado o algoritmo *K-means*. Em [22] os dados de padrão de localização do usuário foram utilizados em uma proposta de sistema biométrico multimodal que uniu às informações do GPS, a estilometria, o uso do aplicativo e o padrão de busca na web.

2.4.6 Fusão do resultados

Fusão é uma abordagem utilizada para combinar informações de múltiplas fontes para melhorar a acurácia ou performance de um método de autenticação biométrica. A informação destas fontes podem ser combinadas de três diferentes formas [12]:

- Fusão a nível de características: envolve a junção de mais de uma característica em uma característica só [12];
- Fusão a nível de *score*: mais de um *score* é combinado em um só para se tomar a decisão [12];
- Fusão a nível de decisão: a fusão é feita combinando decisões (aceito ou rejeitado) feito por múltiplas técnicas de Aprendizado de Máquina usando regras de voto, tais como regras *AND* ou *OR* (Teh et al.(12), 2016 apud Dhage et al.(39), 2015).

Em *frameworks* de autenticação multimodal, uma das questões que precisam de solução é a de como fazer a fusão dos resultados de classificação obtidos para cada um dos padrões que são utilizados. Para resolver esta questão, o trabalho em [22] utilizou a técnica de Centro de Decisão de Fusão, que coleta as n decisões de um detector local e as utiliza para definir se o resultado é -1 ou 1. A abordagem de *score* também pode ser utilizada para a fusão de resultados, conforme se observa em [20], onde é utilizado um centro de decisão que combina todos os *scores* das modalidades, gerando um *score* de decisão global.

2.4.7 Considerações sobre a revisão da literatura

A Tabela 2.9 faz um resumo dos trabalhos de revisão da literatura quanto à verificação estática e dinâmica, em comparação com os aspectos da proposta desenvolvida nessa dissertação.

Tabela 2.9: Resumo comparativo revisão da literatura e *framework* proposto.

Trabalhos	**[7]**	**[8]**	**[36]**	**[37]**	***Framework* Proposto**
Verificação Estática	x	—	x	x	x
Verificação Dinâmica	—	x	—	x	x
Sensores	*Touchscreen*	*Touchscreen*	*Touchscreen*, Acelerômetro, Orientação, Gravidade, Magnetômetro, Giróscópio	*Touchscreen*, Acelerômetro	*Touchscreen*, Acelerômetro, Orientação, Gravidade, Magnetômetro, Giróscópio, Rotação, Aceleração
Dispositivos não Determinados	—	—	x	—	x
Quantidade de Usuários	42	41	95	22	25
Quantidade de Algoritmos	7	2	3	1	7
Melhor Resultado	93,04% acurácia	0% EER intrasessões	96% acurácia	11,5% EER	97,05% F1 *Score*, 1,88% EER, 98,25% Acurácia
Algoritmo do Melhor Resultado	RF	SVM	RF	*Fuzzy*	RF, NBG

Analisando os aspectos resumidos da Tabela 2.9 e na revisão da literatura, para resolver o problema de autenticação apenas baseada em senha, em [7] é proposto um modelo de autenticação baseada em biometria comportamental por verificação estática, no momento de digitação da senha, unindo a senha ao comportamento do usuário ao digitá-la como um fator a mais de autenticação do indivíduo, mas esta abordagem ainda não cobre o momento de interação com aplicação pós *login*. Em [8], foi demonstrado um modelo capaz de autenticar o usuário através dos movimentos de deslizamento horizontal e vertical ao ler uma matéria ou jogar. Estes movimentos também acontecem quando um usuário interage com uma aplicação, utilizando um modelo de autenticação contínua com verficação dinâmica. Em [36] foi proposto um modelo de autenticação baseada em biometria *touch* com foco em aplicações de *Internet Banking*, com verificação apenas estática, e dispositivos não determinados com o melhor resultado em 96%, mas ainda não comtempla a autenticação contínua dinâmica. Em [37], foi proposto um modelo específico para aplicações bancárias que inclui captura de características *touch* tanto para verificação estática quanto dinâmica, mas obteve um EER 11,5%, um valor alto em relação aos outros trabalhos

referenciados.

Baseado nas observações feitas na revisão da literatura resumidas no decorrer deste capítulo, o *framework* proposto neste trabalho, que será detalhado no próximo capítulo, tem o foco em aplicações bancárias *mobile*, assim como em [36] e [37], o qual propõe um modelo que une tanto a autenticação contínua baseada em verificação estática, quanto dinâmica, para autenticação do usuário durante todo o momento de interação com a aplicação, como foi proposto por [37], mas com a expectativa de melhores resultados. Espera-se a geração de resultados próximos ou melhores aos observados em [7], para autenticação ativa estática, e em [8] para autenticação contínua dinâmica. Sendo assim além das camadas de captura no momento de digitação da senha e da interação com as aplicações, será agregado também a captura de informações dos diversos sensores que estão presentes nos dispositivos móveis, mais os sensores de rotação e aceleração, e do padrão de localização do usuário, assim como em [36] e [22] respectivamente.

Para a geração dos modelos serão utilizados seis algoritmos de aprendizado de máquina diferentes, sendo eles: *Naive Bayes Bernoulli* (NBB), *Naive Bayes Gaussian* (NBG), SVM, RF, GB e *Extreme Gradient Boosting* (XGB), conforme justificativas listadas na Tabela 2.10. Apesar de estarem entre os algoritmos mais relevantes [25], que podem ser utilizados para classificação, o algoritmo KNN não será utilizado, pois não apresentou bons resultados quando comparado com RF em [40], e as redes neurais também não serão utilizadas devido a complexidade que possuem.

Tabela 2.10: Justificativas para algoritmos escolhidos.

Algoritmo	**Justificativa**
NBB	Algoritmo simples, e rápido para treino e teste.
NBG	Algoritmo simples, e rápido para treino e teste.
SVM	Está entre os principais listados entre os trabalhos de autenticação baseados em dinâmica *touch* [12], e por ser o que apresentou a menor taxa de erro em [8].
RF	Está entre os principais listados entre os trabalhos de autenticação baseados em dinâmica *touch* [12], e por ser o que possibilitou a melhor acurácia em [36].
GB	É um algoritmo baseado em árvores de decisão que utiliza técnicas de *boosting*, e aumenta a gama de algoritmos que podem tornar possível criar um modelo capaz de autenticar um usuário baseado na dinâmica *touch*.
XGB	É uma evolução do GB, e aumenta a gama de algoritmos que tornam possível criar um modelo capaz de autenticar um usuário baseado na dinâmica *touch*.

Sobre as métricas serão utilizadas a FAR, FRR, EER, REC, PRC, F1 e AC, apesar dos trabalhos publicados durante o desenvolvimento desta dissertação se basearem apenas em acurácia [40] e [41], para esta versão final da proposta que se baseia não apenas em um escopo de auten-

ticação do usuário para verificação estática e dinâmica, como proposto em [41], que considerou apenas um modelo por usuário e utilizou todas as características, mas na junção de outros escopos, ao proposto em [41], sendo considerados também um modelo para vários usuários, com todas as características, um modelo por usuário e um modelo para vários usuários excluindo as características geradas dos outros sensores, levando em consideração apenas as geradas pela interação com o sensor *touchscreen*, e um modelo por usuário e um modelo para vários usuários excluindo as características relacionadas ao tamanho do dedo. Diante destes novos escopos a métrica a ser utilizada será o F1, pois essa é a média harmônica entre REC e PRC, que não sofre forte influência dos TN que são classificados pelo modelo, principalmente em escopos desbalanceados como os que acontecem quando da geração de um modelo com múltiplas classes desbalanceadas, pois neste contexto uma acurácia alta não necessariamente indica um bom desempenho do modelo para a classe, isso pode acontecer nos escopos com um modelo para vários usuários. Sendo mantida a métrica de acurácia apenas para a localização, pois o treino e teste do modelo para a localização não sofreu alteração em relação a [41].

Para a fusão dos resultados dos modelos será utilizada o método utilizado o método de *score*, por ser um método mais simples e intuitivo.

A expectativa é que a junção deste conjunto de características e de sensores em um *framework* de autenticação contínua, proporcione uma melhora de desempenho em relação aos trabalhos já desenvolvidos anteriormente.

3 MODELO PROPOSTO

Neste capítulo será detalhado o *framework* proposto, os módulos que o compõem e os detalhes de implementação.

3.1 MODELO DO *FRAMEWORK* PROPOSTO

O modelo de *framework* proposto tem como objetivo capturar as características dos usuários de interação com uma aplicação *mobile* via *touchscreen* tanto de forma estática, no instante de digitação da senha no *login*, quanto de forma dinâmica, na interação com a aplicação após o *login*. Cada momento possui duas fases: cadastramento e autenticação. Uma visão macro do modelo pode ser vista na Figura 3.1.

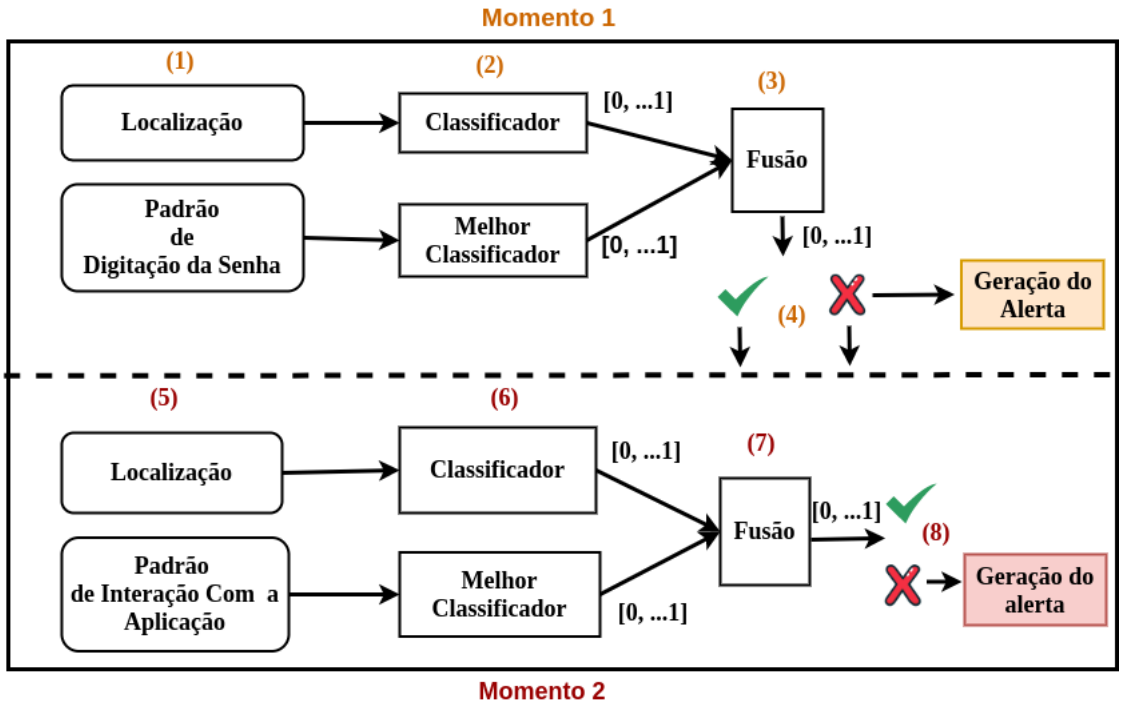

Figura 3.1: Visão macro da arquitetura proposta.

Cada um dos dois momentos de autenticação do usuário é caracterizado de forma distinta:

- **Momento 1** - de digitação da senha, VE;
- **Momento 2** - de interação com a aplicação para a realização de uma transação, VD.

Nas Figuras 3.2 e 3.3, cada um dos momentos é exposto detalhadamente por meio de fluxogramas e, como observados nestes, a adaptação do modelo faz parte da fase de autenticação.

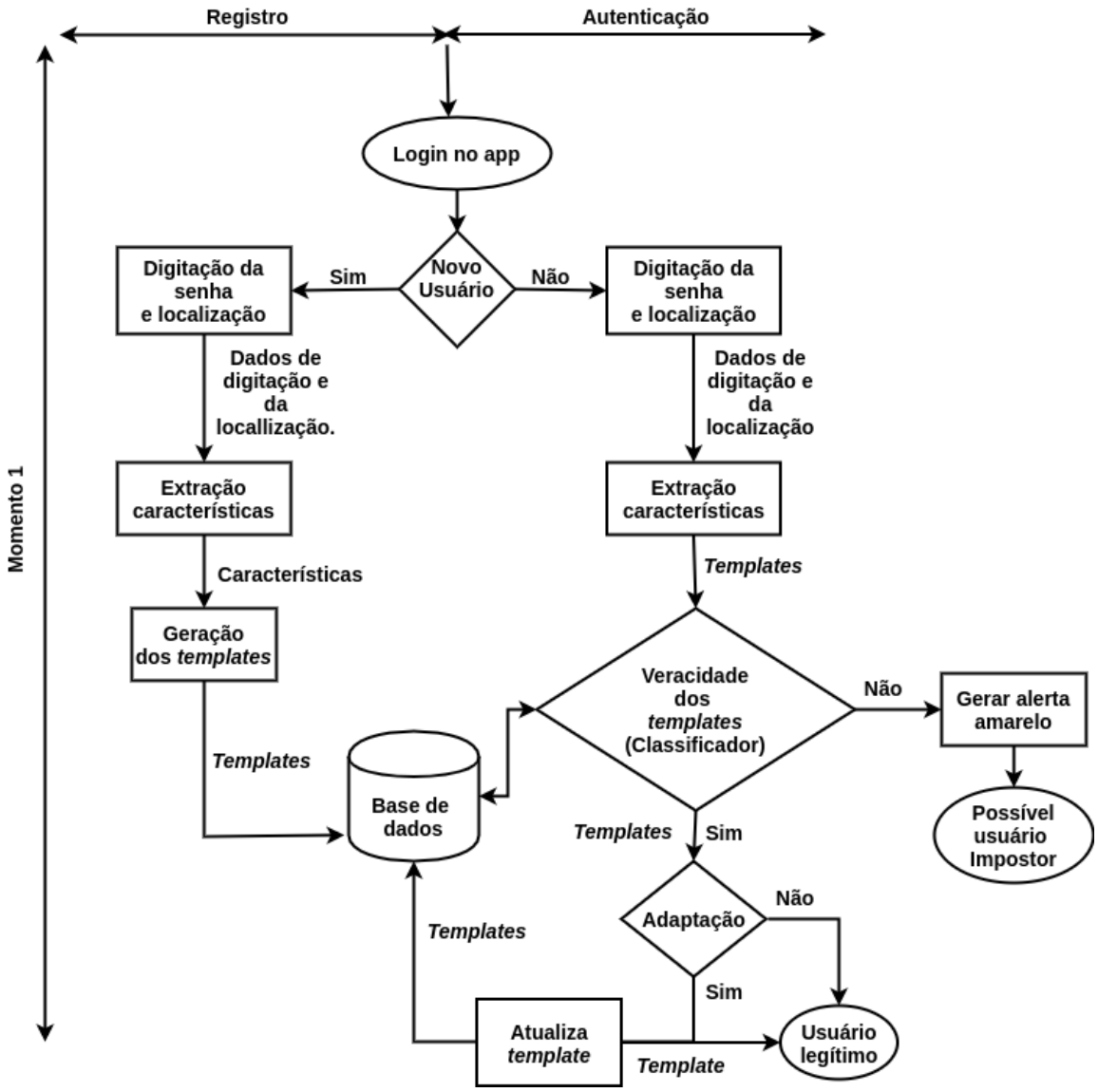

Figura 3.2: Fluxograma Momento 1.

O fluxograma da Figura 3.2 detalha toda a sequência de passos que são implementados para o Momento 1. Para o *login* no aplicativo, o usuário será convidado a cadastrar-se, se este ainda não estiver registrado, se já estiver registrado será inciada a fase de autenticação do usuário, a partir da informação do identificador e da senha já cadastrada anteriormente. O usuário é considerado novo enquanto não tiver a quantidade mínima de *templates* para ser autenticado. Se o *login* for de um usuário novo, será então iniciado o processo de captura das características de digitação da senha para aquele usuário, para que seja criado um modelo para este. O modelo somente é criado a partir do momento que este usuário tenha uma quantidade suficiente de *templates*, 5 (cinco) conforme será detalhado no próximo capítulo, que torne possível a verificação deste. As características coletadas serão salvas em base de dados até que se obtenha a quantidade suficiente. Se for um usuário já identificado na base, este deverá informar identificador e senha, e as características serão coletadas para geração do *template*. Este *template* composto por vetores de características coletados durante a digitação da senha será então utilizado para a verificação estática do usuário.

Se o resultado da autenticação for positivo, e o tempo de atualização do modelo estiver esgotado será feita a adaptação do modelo, a partir da geração de um novo, com base nas últimas interações armazenadas para aquele usuário. Se o resultado da verificação for negativo, será gerado um alerta com indicativo de um possível impostor.

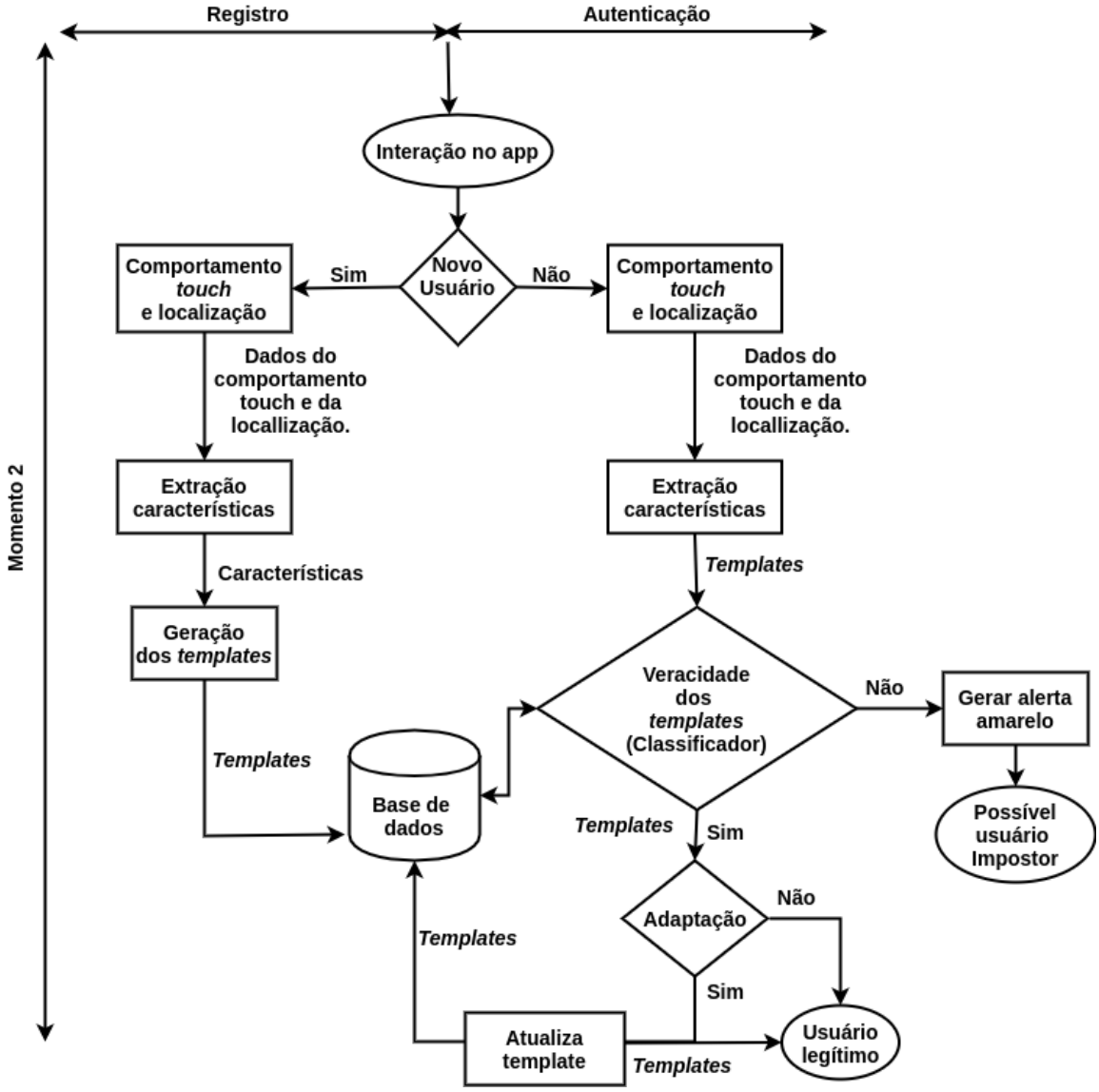

Figura 3.3: Fluxograma Momento 2.

O fluxograma da Figura 3.3 detalha todo a sequência de passos que são implementados para o Momento 2. Para a captura do padrão de interação com a aplicação, assim como no *login* foram desenhados dois caminhos possíveis, o cadastramento se este for um usuário novo ou a verificação dinâmica, caso possua cadastro na base. Se for um usuário novo, será então iniciado o processo de captura das características de interação com a aplicação para aquele usuário, para que seja criado um modelo para este. O modelo somente é criado a partir do momento que este usuário tenha uma quantidade suficiente de *templates*, 15 (cinco) conforme será detalhado no próximo capítulo, permitindo a verificação deste, pois enquanto o usuário não tiver um modelo

criado ele é considerado novo. As características coletadas serão salvas em uma base de dados até que se obtenha a quantidade suficiente para criação do modelo para o usuário. Se for um usuário que possua cadastro na base, as características de interação com a aplicação serão coletadas e, os vetores de características coletados durante esta interação serão então utilizados para a verificação dinâmica do usuário. Se o resultado da autenticação for positivo e o tempo de atualização do modelo estiver esgotado será feita a adaptação do modelo, a partir da geração de um novo com base nas últimas interações armazenadas para aquele usuário. Se o resultado da verificação for negativo, será gerado um alerta com indicativo de um possível impostor.

3.2 IMPLEMENTAÇÃO DO *FRAMEWORK* DE AUTENTICAÇÃO CONTÍNUA

Um *framework* de autenticação contínua é formado de componentes conforme detalhado no Capítulo 2. Nesta seção serão descritos quais os componentes que foram adotados para o modelo de *framework* proposto.

3.2.1 Coleta dos dados

Para a definição do comportamento biométrico na interação com o *touchscreen*, tanto no Momento 1 como no Momento 2, as características foram capturadas por meio de uma aplicação Android, que foi construída com características similares a uma aplicação bancária convencional, publicada na Google *Play Store*, nomeada como Biotouch. A aplicação foi composta de uma tela de cadastro, uma de *login* (L), uma de menu de serviços (MS) e mais 2 telas para cada um dos três serviços disponíveis: a) conta (Cc), uma de menu (Cc1) e uma da transação (Cc2); b) transferência (T), uma de menu (T1) e uma de transação (T2) ; e c) pagamento (P), uma de menu (P1) e uma da transação (P2). O fluxo de cadastro é detalhado na Figura 3.4 e os fluxos de serviços na Figura 3.5.

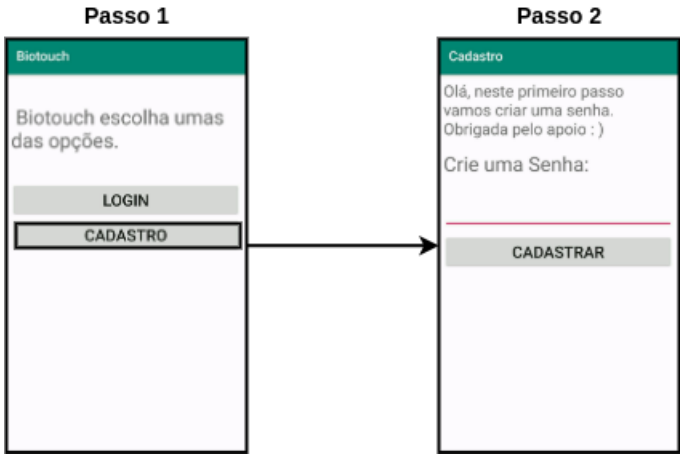

Figura 3.4: Telas aplicativo fluxo cadastro.

Para cadastro na aplicação foi necessário o usuário cadastrar uma senha com 6 a 8 dígitos numéricos. O identificador do usuário foi definido por instalação, de forma transparente, não sendo necessário informar nenhum identificador no momento do cadastro. Na aplicação Biotouch foi utilizado o *Firebase Instance ID*, que fornece um identificador exclusivo para cada instância do aplicativo [42], como identificador do usuário.

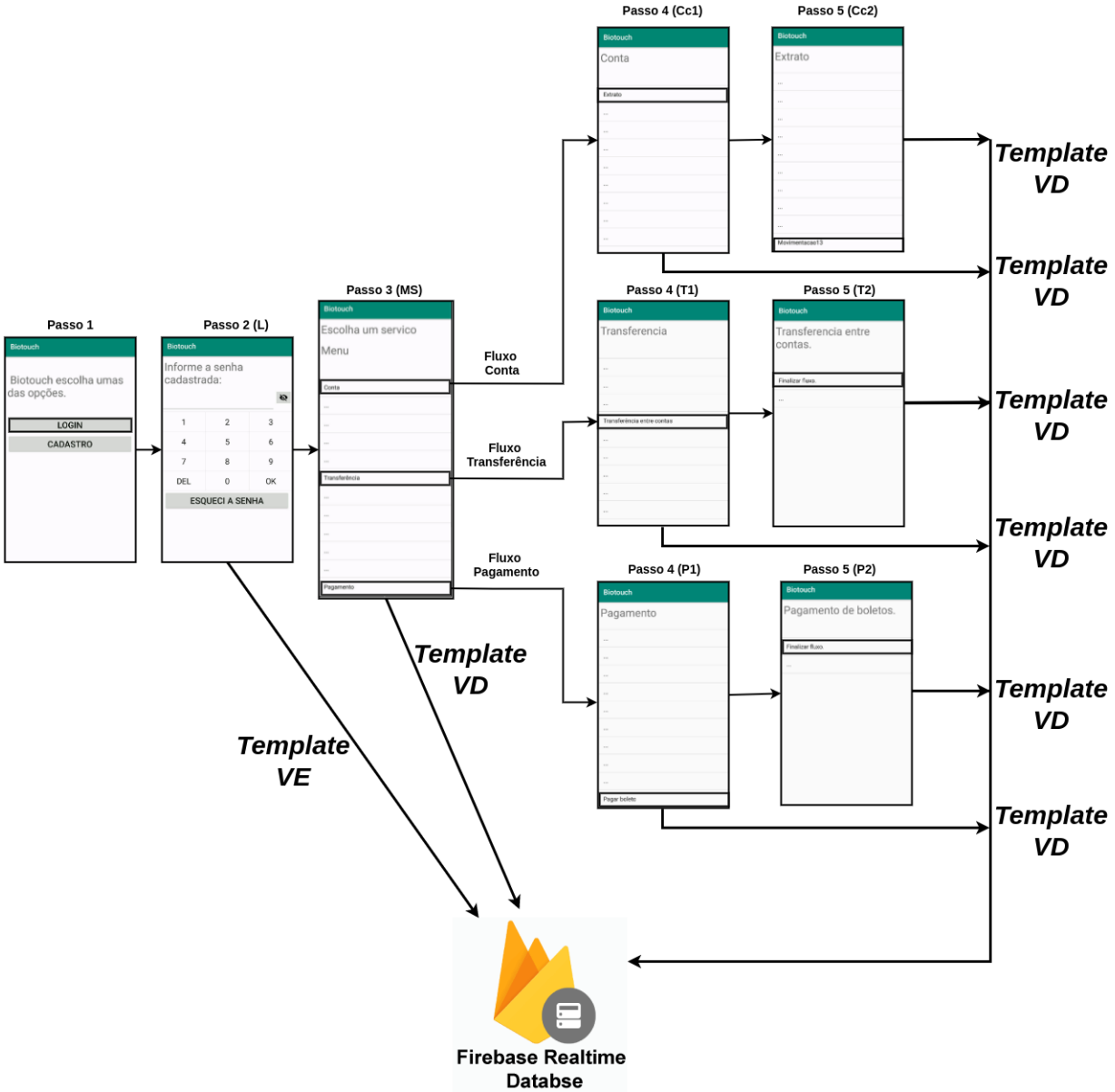

Figura 3.5: Telas aplicativo fluxos.

Para execução de cada fluxo o usuário deveria interagir com 5 telas diferentes para executar todo o fluxo selecionado. Durante a coleta dos dados foi solicitado aos usuários que interagisse pelo menos 5 vezes ao dia com a aplicação durante o período do experimento, 2 semanas, não importando quais dos fluxos fosse executado.

O experimento contou com a participação de 51 voluntários. O período de coleta foi de duas

semanas. A quantidade de *templates* gerados por usuário variou entre 3 e 360, dependendo da quantidade de interações que o usuário teve com a aplicação, pois a cada interação com uma tela foi gerado um *template* para o usuário. Os *templates* foram salvos na plataforma *Firebase*. A Tabela 3.1 detalha a quantidade de *templates* coletada por usuário.

Tabela 3.1: Quantidade de *templates* por usuário.

Usuário	**Quantidade de *templates* por Tela**								
Identificação	**L**	**MS**	**Cc1**	**Cc2**	**T1**	**T2**	**P1**	**P2**	**Total**
c0w2S3pfyeo	1	2	1	–	1	–	–	–	5
c5Ou0JZ3B3s	8	8	2	2	4	4	2	2	32
c8zwbC0QIXc	6	6	2	2	2	2	2	2	24
cBc3b9Cv4X0	17	17	5	5	5	5	7	7	68
cFxPtyX-07w	18	20	8	7	7	7	4	4	75
cK4pQC9AM5g	2	2	1	1	1	1	–	–	8
cLnJdWmHrmE	1	2	1	–	–	–	–	–	5
cQfqJEZZvcE	5	6	2	2	1	1	1	1	19
cTt7L1V7zJ4	2	3	1	–	–	–	–	–	6
cWN_XjnNRDw	14	14	5	5	3	3	6	6	56
cXBqPyLUITc	1	1	1	–	–	–	–	–	3
ceVU4diDpng	2	2	–	–	1	1	1	1	8
cwdNwCqtFAs	11	10	3	3	4	4	3	3	41
d9lTa1CtMYo	3	3	1	1	1	1	1	1	12
dFOtPe4f8Xc	18	22	8	7	7	7	7	7	65
dKAZq_xIG_Q	4	5	2	2	1	1	1	1	17
dUbEbDq40fM	12	12	5	5	3	3	4	4	48
dVGmimRO7bE	20	19	5	5	9	9	5	5	77
dZ98ukA3qYc	4	4	1	1	1	1	2	2	12
dZyNrLCS9AA	9	9	3	3	3	3	3	3	36
dlj7Igoq3HQ	53	54	18	17	17	17	19	19	214
douA39YyKyA	3	3	1	1	1	1	1	1	12
drds94uTXlk	12	8	6	6	1	1	1	1	36
dtfMyro9Zs4	6	6	4	3	2	2	–	–	23
dyRTk2BUAeo	21	19	6	6	6	6	5	5	74
eHC7qNMAdCI	10	11	3	3	5	5	2	2	41
ePCczD0BRSw	6	7	2	2	3	3	2	2	27
eUl_dBC5468	3	5	1	–	1	1	1	1	13
eWI_kICSw_M	6	6	2	2	2	2	2	2	24
eerZKZFi2kY	40	40	12	12	15	15	13	13	160
enJGMKaiFiI	27	28	7	7	13	13	6	6	107
eoWIhgawcZ0	84	85	35	35	26	26	23	23	337

Usuário	Quantidade de *templates* por Tela								
Identificação	**L**	**MS**	**Cc1**	**Cc2**	**T1**	**T2**	**P1**	**P2**	**Total**
eqmPzjjzrHk	23	25	5	4	11	10	9	9	96
ev9fChXnR3I	17	18	11	11	2	2	5	5	69
f0ttzpoZyeA	75	75	13	13	52	52	10	10	300
f41VjsojN_E	8	10	2	2	4	4	2	2	34
fDAVsmA3HUY	69	70	22	22	24	24	24	24	279
fGTU-LDm8uM	20	20	7	7	6	6	7	7	80
fIewI06H8u8	31	30	10	10	10	10	10	10	121
fM-UHv6MXeo	1	2	1	1	1	1	–	—	7
fRA_pBm0ks4	18	18	6	6	6	6	6	6	72
fUB30EtiU0Q	12	12	4	4	5	5	3	3	48
f_sQU3hUQEs	3	3	1	1	1	1	1	1	12
fc-Dwcad0wU	7	8	4	3	2	2	2	2	30
fc9wWdmmcBc	5	5	2	2	2	2	1	1	20
fdMwZ2D5l5w	6	6	2	2	2	2	2	2	24
ffdzWINClJ4	14	16	6	4	6	6	4	4	60
fhw9jzhvkGs	90	90	38	38	49	49	3	3	360
fmVXDwdw20Q	25	21	7	7	7	7	7	7	88
fnCdC1RnI38	9	10	6	5	2	2	2	2	38
fuVCn-BNrJM	5	5	2	2	1	1	2	2	20

Neste trabalho cada *template* é representado pelos eventos gerados a partir de toques efetuados durante uma sessão de interação com cada tela. E para a autenticação é utilizado o conjunto de *templates* gerados durante a sessão. Sendo assim, a quantidade de vetores de informações utilizados a cada autenticação pode variar.

Os modelos de *smartphones* utilizados no experimento foram: SM-G973F, SM-G9600, LG-M250, Moto G (4), SM-A305GT, SM-G9650, SM-G970F, SM-G975F, ASUS X00QD, F670S, GM1900, GT-I9500, LGM-M700, MI 8, Mi 9T, Mi A2, Mi A3, Moto E(4), Moto X4, Moto Z2 Play, MotoG3, One Vision, POCOPHONE F1, Redmi 7, Redmi Note 4, Redmi Pro, SM-A530F, SM-G530H, SM-G570M, SM-G930F, SM-G935F, SM-G955F, SM-G955U, SM-N950F, SM-N9600, SM-N970F, SM-N975F, X00HD, X00LD, XT1635-02, XT1710-02, conforme informações coletadas via plataforma Firebase [43].

As versões do Android nos *smartphones* foram: 5.0.1,5.0.2, 6.0, 6.0.1, 7.0, 7.1.1, 8.0.0, 8.1.0, 9 e 10, informações coletadas na plataforma *Firebase*.

3.2.2 Características capturadas

Para o *framework* foram capturadas 31 características no total, mas destas apenas 29 foram utilizadas no Momento 1, pois o armazenamento das coordenadas de digitação pode gerar a dedução da senha. Para o Momento 2 todas as 31 características foram utilizadas. Todas as caraterísticas coletadas são detalhadas na Tabela 3.2.

Tabela 3.2: Características coletadas em relação aos sensores e os Momentos 1 e 2

Característica	**Sensor**	**Momento**
Tempo Pressiona Pressiona	*Touchscreen*	1,2
Tempo Pressiona Solta	*Touchscreen*	1,2
Tempo Solta Pressiona	*Touchscreen*	1,2
Tempo Solta Solta	*Touchscreen*	1,2
Média do tempo de pressionamento	*Touchscreen*	1,2
Pressão	*Touchscreen*	1,2
Media da pressão	*Touchscreen*	1,2
Tamanho do dedo	*Touchscreen*	1,2
Media de tamanho do dedo	*Touchscreen*	1,2
Coordenada x	*Touchscreen*	2
Coordenada y	*Touchscreen*	2
Força de aceleração ao longo do eixo X (incluindo a gravidade) [15]	Acelerômetro	1,2
Força de aceleração ao longo do eixo Y (incluindo a gravidade) [15].	Acelerômetro	1,2
Força de aceleração ao longo do eixo Z (incluindo a gravidade) [15].	Acelerômetro	1,2
Taxa de rotação ao redor do eixo X [15].	Giroscópio	1,2
Taxa de rotação ao redor do eixo Y [15].	Giroscópio	1,2
Taxa de rotação ao redor do eixo Z [15].	Giroscópio	1,2
Campo geomagnético do ambiente para o eixo físico X em T	Magnetômetro	1,2

Característica	Sensor	Momento
Campo geomagnético do ambiente para o eixo físico Y em T	Magnetômetro	1,2
Campo geomagnético do ambiente para o eixo físico Z em T	Magnetômetro	1,2
Componente do vetor de rotação ao longo do eixo X (X* sen (/ 2)) [15].	Sensores de rotação (software ou hardware)	1,2
Componente do vetor de rotação ao longo do eixo Y (Y* sen (/ 2)) [15].	Sensores de rotação (software ou hardware)	1,2
Componente do vetor de rotação ao longo do eixo Z (Z* sen (/ 2)) [15].	Sensores de rotação (software ou hardware)	1,2
Componente escalar do vetorde rotação ((cos (/ 2)) [15].	Sensores de rotação (software ou hardware)	1,2
Acurácia estimada	Sensores de rotação (software ou hardware)	1,2
Força de aceleração ao longo do eixo X (excluindo a gravidade) [15].	Sensores de aceleração (software ou hardware)	1,2
Força de aceleração ao longo do eixo Y (excluindo a gravidade) [15].	Sensores de aceleração (software ou hardware)	1,2
Força de aceleração ao longo do eixo Z (excluindo a gravidade) [15].	Sensores de aceleração (software ou hardware)	1,2
Força da gravidade ao longo do eixo X[15].	Sensores de gravidade (software ou hardware)	1,2
Força da gravidade ao longo do eixo Y [15].	Sensores de gravidade (software ou hardware)	1,2
Força da gravidade ao longo do eixo Z [15].	Sensores de gravidade (software ou hardware)	1,2

A representação dos eixos X, Y e Z em um *smartphone* é detalhada na Figura 3.6, visando facilitar o entendimento sobre a coleta das características em relação ao eixos conforme detalhada na Tabela 3.2.

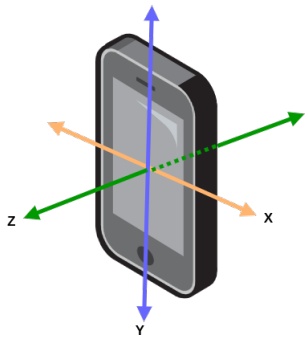

Figura 3.6: Representação eixos *smartphone*.

3.2.3 Criação dos modelos

Para criação dos modelos que serão utilizados em VE e VD é proposto que sejam utilizados 6 algoritmos de Aprendizado de Máquina diferentes para encontrar o que terá F1 de pelo menos 90% e FAR de no máximo 10%, quando comparado com os dados de todos os outros usuários participantes do experimento, para o padrão do usuário nos Momentos 1 e 2, sendo estes: NBB (*Naive Bayes Bernoulli*); NBG (*Naive Bayes Gaussian*); SVM (*Support Vector Machine*); RF (*Random Forest*); GB (*Gradient Boosting*); XGB (*Extreme Gradient Boosting*).

As ferramentas utilizadas para a construção dos modelos foram a biblioteca *scikit-learn*, versão 0.20.3, e a *XGBoost Python*, versão 0.90.

3.2.3.1 Algoritmo NBB e NBG

Naive Bayes é um classificador probabilístico que pressupõe uma independência entre os dados [44] [45]. Em outras palavras, dado o valor da classe de uma instância a probabilidade de observar a conjunção $a_1, a_2 ... a_n$ é apenas o produto das probabilidades para os atributos individuais[45]. A aprendizagem *bayesiana* simplesmente calcula a probabilidade de cada hipótese, considerando-se os dados, e faz previsões de acordo com ela. Isto é, as previsões são feitas com o uso de todas as hipóteses, ponderadas por suas probabilidades. Desse modo, a aprendizagem é reduzida à inferência probabilística. [46].

Uma vantagem deste modelo é que por supor uma independência entre os atributos ele é capaz de fazer o treinamento de um modelo com um número pequeno de amostras. Esse modelo simples também pode trabalhar com dados que tem vários atributos. No caso do *Naive Bayes Bernoulli* é suposto atributos booleanos discretos e para o *Naive Bayes Gaussian* é suposto uma distribuição Gaussiana [44].

Na biblioteca *scikit-learn* para o algoritmo NBB foi utilizada a classe *BernoulliNB* e para o NBG foi utilizada a classe *GaussianNB*, ambas do pacote *sklearn.naive_bayes* com os parâmetros padrão.

3.2.3.2 Algoritmo SVM

Uma SVM, Máquina de Vetores de Suporte, é um algoritmo que tenta fazer a adequação de uma linha (ou plano ou hiperplano) entre as diferentes classes de modo a maximizar a distância da linha até os pontos das classes. Dessa maneira, ela tenta encontrar uma separação robusta entre as classes. Os vetores de suporte são os pontos da fronteira do hiperplano divisor [44]. A SVM é uma técnica discriminante e sempre retorna o mesmo parâmetro ideal de hiperplano [47].

As três propriedades que tornam as SVMs atraentes são [46]:

- Constroem um separador de margem máxima, o que as ajuda a generalizar bem [46] sobre os dados a serem classificados;
- Criam uma separação linear no hiperplano, mas têm a capacidade de incorporar os dados em um espaço de dimensão superior, usando assim o chamado truque do *kernel*, fazendo com que o espaço de hipóteses seja expandido em relação aos métodos que usam representações estritamente lineares [46]. Como pode ser notado na Figura 3.7 dependendo do *kernel* é possível ter uma classificação diferente das classes, SVC é a classe no *scikit-learn* para a implementação da SVM. Nesse exemplo, que pode ser encontrado na documentação da biblioteca *scikit-learn* [48], é utilizado o *dataset* irís, um *dataset* clássico para exemplos de classificação. As classes, que neste caso são tipos de flores: setosa, versicolor e virginica e os dados das classes são visualmente expostos pelas cores azul escuro, azul claro e vermelho. Conforme é possível notar na Figura 3.7 dependendo do tipo de *kernel* a classificação dos dados e as fronteiras das classes são alteradas;

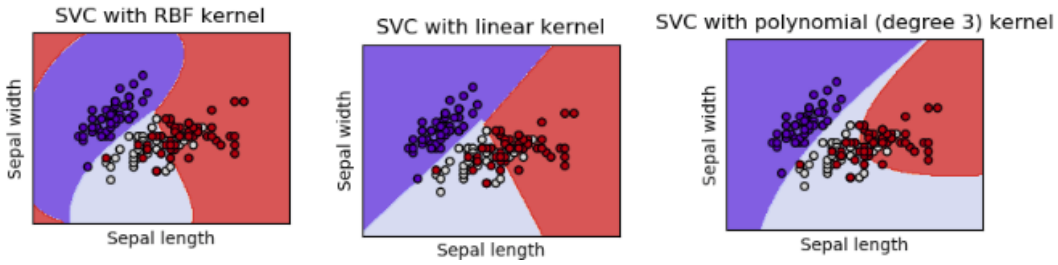

Figura 3.7: Representações *kernel* SVM classificação [48].

- São um método não paramétrico, elas mantêm exemplos de treinamento e podem precisar de todos eles. Por outro lado na prática acabam mantendo apenas uma pequena fração do número de exemplos. Assim as SVMs combinam as vantagens de métodos paramétricos e não paramétricos: elas têm a flexibilidade para representar funções complexas, mas são resistentes à superadequação [46].

Na biblioteca *scikit-learn* para o algoritmo SVM foi utilizada a classe *SVC*, do pacote *sklearn.svm* com os seguintes hiperparâmetros customizados [49]:

- *kernel*: especifica o tipo de kernel a ser usado no algoritmo. Ele deve ser um dos itens '*linear*', '*poli*', '*rbf*', '*sigmoid*', '*precomputed*'. Se nada for fornecido, '*rbf*' será usado ;
- *gamma*: indica o coeficiente do *kernel* para '*rbf*', '*poly*' e '*sigmoid*';
- *C*: parâmetro de regularização. A força da regularização é inversamente proporcional a *C*. Deve ser estritamente positiva;
- *class_weigh*t: indica o peso da classe. Se não for dada, todas as classes terão peso um;
- *degree*: indica o grau da função polinomial do *kernel* ('*poli*'). É gnorado por todos os outros *kernels*.

3.2.3.3 Algoritmo RF

Aprendizagem em árvore de decisão é um método para a aproximação de funções de valor discreto, no qual cada função aprendida é representada por uma árvore de decisão. A árvore também pode ser representada com um conjunto de regras se-então [45]. Uma árvore de decisão usa uma estrutura de árvore para representar um número de possíveis caminhos de decisão e um resultado para cada caminho [24], estas classificam as instâncias da raiz para algum nó folha [45].

Random Forest é uma abordagem de aprendizagem *ensemble* para classificação, na qual "alunos fracos" colaboram para formar "aprendizes fortes", usando uma grande coleção de árvores de decisão relacionadas (a Floresta de aleatória)(Awad e Khanna(47), 2015 apud Breiman(50), 2001). Uma floresta aleatória (*Random Forest*) é um conjunto de árvores de decisões. Cada árvore em uma floresta aleatória é construída de uma seleção "aleatória", ou subgrupo, de exemplos de treino e atributos [51]. A metodologia *bagging* é utilizada para corrigir a tendência das árvores de decisão à superadequação. Ao criar várias árvores treinadas com subamostras e atributos aleatórios dos dados, a variância é reduzida [44]. *Baggging*, ou *Bootstrap aggregating*, se caracteriza pela utilização de múltiplos modelos, no caso do RF árvores, a partir de diferentes conjuntos dos dados de treino.

O princípio geral dos métodos *ensemble* é construir uma combinação linear de alguns métodos para ajuste de modelos, ao invés de utilizar um método de ajuste único [52]. *Ensemble* (de conjunto) métodos empregam múltiplos "aprendizes" para solucionar um problema com uma melhor habilidade de generalização e com uma predição mais acurada [47].

Na biblioteca *scikit-learn* para o algoritmo RF foi utilizada a classe *RandomForestClassifier*, do pacote *sklearn.ensemble* com os seguintes hiperparâmetros customizados:

- *random_state*: controla a aleatoriedade do *bootstrapping* das amostras usadas ao construir árvores (se *bootstrap* = True) e a amostragem dos recursos a serem considerados ao procurar a melhor divisão em cada nó (se max_features <n_features) [53];
- *n_jobs*: indica a quantidade de cores de CPU a ser utilizada no computador;

- *n_estimators*: indica o número de árvores na floresta [53];
- *max_depth*: indica a profundidade máxima da árvore. Se *None*, os nós serão expandidos até que todas as folhas estejam puras ou até que todas as folhas contenham o mínimo de amostras min_samples_split [53];
- *max_features*: indica o número de características a serem considerados ao procurar a melhor divisão [53];
- *min_samples_leaf*: indica o número de características a serem considerados ao procurar o número mínimo de amostras necessárias para estar em um nó folha [53].
- *min_samples_split*: indica o número mínimo de amostras necessárias para dividir um nó interno [53];
- *bootstrap*: indica se as amostras de autoinicialização são usadas na construção de árvores. Se *False*, o conjunto de dados inteiro será usado para criar cada árvore [53].
- *criterion*: indica a função para medir a qualidade de uma divisão. Os critérios suportados são "*gini*" para a impureza de *Gini* e "entropia" para o ganho de informações [53].
- *class_weight*: indica os pesos associados às classes no formato *class_label: weight*. Se não for atribuída, todas as classes terão peso um [53].

3.2.3.4 Algoritmo GB

Gradient Boosting é um classificador *ensemble* [54], baseado em *boosting* em árvores de decisão, ou seja durante a construção dos múltiplos modelos, árvores, cada modelo criado aprende a fazer a correção dos erros gerados pelo modelo anterior, dentro da sequência de modelos criados. Em *Gradient Boosting Machines*, o procedimento de aprendizado se ajusta consecutivamente a novos modelos para fornecer uma estimativa mais precisa da variável de resposta. A ideia principal por trás desse algoritmo é construir os novos aprendizes base para serem maximamente correlacionados com o gradiente negativo da função de perda, associado a todo o conjunto. As funções de perda aplicadas podem ser arbitrárias [55].

Na biblioteca *scikit-learn* para o algoritmo GB foi utilizada a classe *GradientBoostingClassifier*, do pacote *sklearn.ensemble* com os seguintes hiperparâmetros customizados [56]:

- *random_state*: controla a semente aleatória fornecida a cada estimador de árvore a cada iteração de *Boosting*. Além disso, controla a permutação aleatória dos recursos em cada divisão;
- *n_estimators*: indica o número de estimadores selecionados por parada antecipada (se *n_iter _no_change* for especificado). Caso contrário, é definido como *n_estimators*;

- *learning_rate*: indica a redução da contribuição de cada árvore por *learning_rate*. Há uma troca entre *learning_rate* e *n_estimators*;
- *max_depth*: indica a profundidade máxima dos estimadores de regressão individuais. A profundidade máxima limita o número de nós na árvore. Ajuste esse parâmetro para obter o melhor desempenho; o melhor valor depende da interação das variáveis de entrada;
- *min_samples_split*: indica o número mínimo de amostras necessárias para dividir um nó interno;
- *min_samples_leaf*: indica o número mínimo de amostras necessárias para estar em um nó folha. Um ponto de divisão em qualquer profundidade só será considerado se deixar pelo menos amostras de treinamento *min_samples_leaf* em cada um dos ramos esquerdo e direito. Isso pode ter o efeito de suavizar o modelo, especialmente em regressão ;
- *max_features*: indica o número de características a serem considerados ao procurar a melhor divisão.

3.2.3.5 Algoritmo XGB

O XGBoost é um sistema de aprendizado de máquina escalável para *boosting* de árvores [57], ele cria uma arvore fraca e, então, "melhora" as árvores subsequentes (faz um *boosting*) a fim de reduzir os erros residuais. O algoritmo tenta capturar e tratar qualquer padrão de erros, até que pareçam ser aleatórios[44]. O XGBoost é uma implementação específica de *Gradient Boosting* de forma aprimorada em termos de performance e utilização de recursos.

Na biblioteca *XGBoost Python* para o algoritmo GB foi utilizada a classe *XGBClassifier*, do pacote *xgb* com os seguintes hiperparâmetros customizados:

- *n_estimators*: indica o número de rodadas ou árvores melhoradas [44].
- *colsample_bytree*: indica a taxa de subamostra de colunas ao construir cada árvore. A subamostragem ocorre uma vez para cada árvore construída [58].
- *max_depth*: indica a profundidade máxima de uma árvore [58].
- *reg_alpha*: indica o termo de regularização *L1* em pesos [58], termo responsável pela punição dos termos menos preditivos nas árvores.
- *reg_lambda*: indica o termo de regularização *L2* em pesos [58], termo rensponsável pela punição de folhas grandes nas árvores.
- *subsample*: indica a proporção de subamostra das instâncias de treinamento [58].

Como o NBB e o NBG são algoritmos mais simples e bem mais rápidos para predição e treino, por serem baseados em probabilidades, estes tem preferência em relação aos outros algoritmos

que são baseados em métodos *ensemble*, se atingirem o F1 definido com limiar em 90% para este *framework* proposto.

Para ajuste dos hiperparâmetros dos algoritmos baseados em métodos *ensemble* foi utilizada a técnica de *Grid Search*, utilizando uma lista de parâmetros pré-definida para cada um dos algoritmos SVM, RF, GB e XGB, conforme tabela 3.3.

Tabela 3.3: Hiperparâmetros utilizados para os algoritmos baseado em métodos *ensemble*.

Algoritmo	**Parâmetros**
SVM	{'kernel': ['rbf'], 'gamma': ['scale', 'auto', 1e-2, 1e-3, 1e-4], 'C': [0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.1], 'class_weight': [{0:1,1:2}, 'balanced']}, {'kernel': ['linear'], 'C': [0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.1], 'class_weight': [{0:1,1:2}, 'balanced']}, {'kernel': ['poly'], 'C': [0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.1], 'class_weight': [{0:1,1:2}, 'balanced'], 'degree': [3,5]}
RF	{'random_state': 0, 'n_jobs': 2, 'n_estimators': [20, 25, 30], 'max_depth': [3, 5, None], 'max_features': [1, 3, 5, 'auto'], 'min_samples_leaf': [0.3, 0.4, 0.5], 'min_samples_split': [0.3, 0.4, 0.5, 6], 'bootstrap': [True, False], 'criterion': ['gini', 'entropy'], 'class_weight': [{0:1,1:2}, 'balanced']}
GB	{'n_estimators': [10, 20, 30, 75, 100], 'learning_rate': [0.001, 0.01, 0.1], 'max_depth': [5, 6, 7], 'min_samples_split': [0.3, 0.4, 0.45, 0.5], 'min_samples_leaf': [0.20, 0.25, 0.3, 0.4], 'max_features': [3, 7, 10, 20, None]}
XGB	{'n_estimators': [20, 30, 40, 100], 'colsample_bytree': [0.6, 0.7, 0.8], 'max_depth': [15,20,25], 'reg_alpha': [1.1, 1.2, 1.3], 'reg_lambda': [1.1, 1.2, 1.3], 'subsample': [0.7, 0.8, 0.9]}

No total cada usuário teve 3 modelos criados, sendo um para a verificação estática (*login*), um para a verificação dinâmica (interação com a aplicação após *login*), e um para o padrão de localização, que foi compartilhado entre os Momentos 1 e 2. Os modelos dos Momentos 1 e 2 podem ser individuais ou compartilhados com outros usuários, isso irá depender de qual será o melhor escopo para encontrar o F1 pretendido para o usuário, a partir de 90%, para cada modelo e do comportamento do modelo quando confrontado com os *templates* de outros usuários quanto

ao FAR de no máximo 10%. Os escopos possíveis são detalhados no Capítulo 4.

3.2.4 Adaptação dos modelos

A fase de adaptação do modelo se refere ao período de tempo em que se torna necessária a atualização do modelo criado para um indivíduo, baseado no fato de que o padrão de interação com a aplicação de um usuário pode ser alterado ao longo do tempo, devido a algum aspecto físico ou de coordenação motora que foi alterado, por exemplo, ou devido à alteração da informação atrelada aquele modelo, no caso do modelo ligado ao padrão de digitação da senha.

O experimento foi conduzido durante 2 semanas, sendo este o tempo máximo para adaptação do modelo que pode ser considerado diante do cenário de experimento realizado. Para definições sobre um tempo maior de adaptação seria necessário conduzir uma pesquisa com captura de dados por um maior período de tempo que poderia ser: semanas, meses, trimestres, semestres ou ano. O que foge do escopo desta pesquisa, podendo ser desenvolvido em um trabalho futuro.

3.2.5 Características mais importantes

No *framework* é proposta a captura de dados de vários sensores disponíveis no dispositivo, além das informações do movimento na interação *touch*, capturadas via sensor *touchscreen*. Para entender qual a representação de importância de cada uma destas características foi utilizado o algoritmo RF com os parâmetros definidos na Tabela 3.4, tanto para a verificação estática quanto dinâmica para gerar os *rankings* das características mais importantes. Os parâmetros foram definidos de forma empírica, com base nos melhores resultados obtidos em tempo de treino e teste.

Tabela 3.4: Hiperparâmetros do algoritmo RF para identificação de características mais importantes.

Hiperparâmetros para algoritmo RF
{'n_estimators': 40, 'n_jobs': 2, 'random_state': 0, 'bootstrap': False, 'criterion': 'entropy', 'max_depth': 5, 'max_features': 9, 'min_samples_leaf': 3}

Para a definição das características mais importantes nos Momentos 1 e 2, excluídas as de localização, foram criados modelos únicos para extração destas características com uma visão global.

3.2.6 Localização

O padrão de localização de um usuário define um contexto que geralmente é mantido de forma padronizada para cada indivíduo. E pode ser uma informação de contexto importante quando se estabelece padrões de autenticação, pois pode ser uma fonte a mais de verificação da identidade de um usuário. Com base nessas informações, neste trabalho para a análise do padrão de localização

dos usuários foi utilizado o algoritmo SVM *one-Class* para a definição do modelo de localização, com a expectativa de encontrar uma acurácia de pelo menos 90%, com parâmetros definidos conforme Tabela 3.5. Os parâmetros foram definidos de forma empírica, com base nos melhores resultados obtidos em tempo de treino e teste. Na biblioteca *scikit-learn* para o algoritmo SVM *one-Class* foi utilizada a classe *OneClassSVM*, do pacote *sklearn.svm*.

Tabela 3.5: Hiperparâmetros do algoritmo SVM *One-Class* para definição do modelo da localização do usuário.

Hiperparâmetros para algoritmo SVM *One-Class*
nu=0,7, kernel="poly", gamma=0.09

3.2.7 Fusão dos resultados de predição dos modelos

Para o modelo proposto a regra de fusão dos resultados dos classificadores é baseada em *score*, utilizando a média entre a acurácia para a localização e o F1 para os Momentos 1 e 2, dos classificadores conforme mostrado na Figura 3.8. A pesar de em (41) ter sido proposto a dedução do desvio padrão do valor da média, esse requisito foi removido nesta proposta de versão final, pois nesta versão o modelo criado também já passará por um crivo de teste contra todos os outros usuários do sistema, como será explicado na próxima seção, fazendo com que não seja mais necessária a penalização gerada no *score* final pela dedução do desvio padrão, e visto que todos os valores de *score* somente serão aceitos se tiverem com valor a partir de 90% tanto para a localização e quanto para os Momentos 1 e 2. Nos Momentos 1 e 2 será utilizada o F1 do melhor classificador, que poderá ser individual do usuário ou compartilhada com outros, sempre considerando o F1 para a classe.

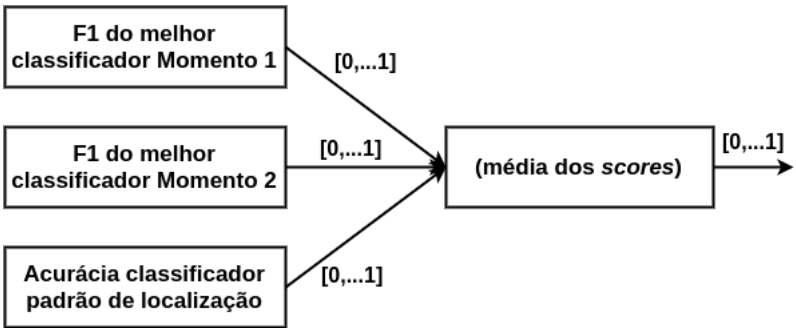

Figura 3.8: Diagrama de fusão dos resultados.

4 RESULTADOS E ANÁLISES

Nesta seção serão detalhados os resultados dos experimentos definidos para validação do modelo proposto visando encontrar os melhores algoritmos para alcançar o limiar de F1 definido em 90%, valor definido com base na média dos valores de acurácia referenciados em [7] e [36], 82,53%, 93,04% e 96%. Como o *framework* utiliza escopos heterogêneos a acurácia foi substituida pelo F1.

4.1 CENÁRIOS DOS EXPERIMENTOS

Para definir qual a quantidade mínima de interações necessárias com a aplicação para a criação de um modelo de Aprendizado de Máquina, que seja capaz de obter 90% de F1 e FAR de no máximo 10%, quando o modelo de cada usuário for confrontado com os *templates* de outros usuários, foram definidos três cenários: C1, C2 e C3, conforme Tabela 4.1. É usada uma proporção de um *template* de *login* para três de interação com a aplicação nos cenários de teste, pois para finalizar um fluxo o usuário deve interagir com três telas, conforme detalhado no Capítulo 3, após o *login*: (1) Tela Menu Serviços, (2) Tela Menu da Transação e (3) Tela da Transação.

Tabela 4.1: Detalhamento dos cenários de experimento.

Tipo	**Quantidade de *Templates***	**Quantidade de *Templates* de Treino**	**Quantidade de *Templates* de Teste**	**Cenário**
VE	A partir de 10	5	A partir de 5	1
VD	A partir de 30	15	A partir de 15	1
VE	A partir de 15	10	A partir de 5	2
VD	A partir de 45	30	A partir de 15	2
VE	A partir de 20	15	A partir de 5	3
VD	A partir de 60	45	A partir de 15	3

Para estes cenários, dos 51 usuários que instalaram a aplicação, apenas 25 participaram do experimento, como usuários legítimos, pois somente estes geraram a quantidade mínima de *templates* para C1: 10 interações com o *login* e pelo menos 10 interações completas nos fluxos das transações, representando 30 *templates* pós login. Os dados dos outros 26 usuários, que não disponibilizaram a quantidade mínima de *templates* necessários para o cenário 1, foram utilizados para gerar dados de impostores para treino e testes, e também foram utilizadas para o cálculo do FAR sobre impostores. A Tabela 4.2 detalha os 25 usuários, e a quantidade de *templates* que cada um gerou durante o período do experimento.

Tabela 4.2: Quantidade de *templates* dos 25 usuários legítimos participantes do experimento.

Usuário	**Quantidade de *Templates* por Tela**								
Identificação	**L**	**MS**	**Cc1**	**Cc2**	**T1**	**T2**	**P1**	**P2**	**Total**
drds94uTXlk	12	8	6	6	1	1	1	1	36
cwdNwCqtFAs	11	10	3	3	4	4	3	3	41
eHC7qNMAdCI	10	11	3	3	5	5	2	2	41
dUbEbDq40fM	12	12	5	5	3	3	4	4	48
fUB30EtiU0Q	12	12	4	4	5	5	3	3	48
cWN_XjnNRDw	14	14	5	5	3	3	6	6	56
ffdzWINClJ4	14	16	6	4	6	6	4	4	60
dFOtPe4f8Xc	18	22	8	7	7	7	7	7	65
cBc3b9Cv4X0	17	17	5	5	5	5	7	7	68
ev9fChXnR3I	17	18	11	11	2	2	5	5	69
fRA_pBm0ks4	18	18	6	6	6	6	6	6	72
dyRTk2BUAeo	21	19	6	6	6	6	5	5	74
cFxPtyX-07w	18	20	8	7	7	7	4	4	75
dVGmimRO7bE	20	19	5	5	9	9	5	5	77
fGTU-LDm8uM	20	20	7	7	6	6	7	7	80
fmVXDwdw20Q	25	21	7	7	7	7	7	7	88
eqmPzjjzrHk	23	25	5	4	11	10	9	9	96
enJGMKaiFiI	27	26	7	7	13	13	6	6	107
fIewI06H8u8	31	30	10	10	10	10	10	10	121
eerZKZFi2kY	40	40	12	12	15	15	13	13	160
dlj7Igoq3HQ	53	54	18	17	17	17	19	19	214
fDAVsmA3HUY	69	70	22	22	24	24	24	24	279
f0ttzpoZyeA	75	75	13	13	52	52	10	10	300
eoWIhgawcZ0	84	85	35	35	26	26	23	23	337
fhw9jzhvkGs	90	90	38	38	49	49	3	3	360

Baseado no número máximo de *templates* gerados pelos 25 usuários legítimos, foi definido o número de usuários que iriam participar de cada cenário e de cada Momento conforme Tabela 4.3.

Tabela 4.3: Quantidade de participantes por cenário VE e VD.

Participantes VE e VD			
	C1	**C2**	**C3**
VE	25 de 25	18 de 25	13 de 25
VD	23 de 23	18 de 23	11 de 23

4.2 VERIFICAÇÃO ESTÁTICA E DINÂMICA

Para a definição do modelo de acordo com as regras do *framework* são seguidos os seguintes passos:

1. Todos os 6 algoritmos são treinados e testados, com dados balanceados, é utilizada a mesma quantidade de vetores, linhas contidas nos *templates*, para o usuário legítimo e ilegítimo, determinada pela quantidade de vetores contida nos *templates* do usuário legítimo;
2. São identificados os algoritmos que obtiveram F1 a partir de 90%;
3. São então descartados aqueles modelos que obtiverem acurácia de 100%, pois este comportamento pode indicar *overfitting*, ou que os dados ainda não são suficientes para que seja definido o padrão do usuário;
4. Se entre os modelos com F1 a partir de 90% estiver o NBB ou o NBG, estes terão preferência, pois são algoritmos mais simples e rápidos para predição. Caso contrário, o modelo com maior valor de F1 é selecionado;
5. O melhor modelo selecionado no passo anterior, será então confrontado com os dados de pelo menos outros 50 usuários, todos os outros usuários que participaram do experimento, em um ambiente real esse valor pode ser maior. O modelo somente será considerado bom, se obtiver um FAR de no máximo 10%, que nesse caso pode representar que entre os 50 outros usuários impostores, 5 tem um padrão de comportamento que foi identificado como similar ao do usuário avaliado, com base nas características utilizadas no experimento.

No treino a quantidade de *templates* foi definida conforme os cenários e no teste, foi realizada a autenticação de todos os outros *templates* do usuário deduzidos os utilizados no treino. Tanto no treino quanto para testes foram utilizados conjunto de dados das classes de forma equilibrada, com a proporção de 50% de dados de usuários legítimos e 50% de dados de usuários ilegítimos, isto se aplica para os escopos de um modelo por usuário, pois nos modelos multiclasse a quantidade de vetores, linhas contidas nos *templates*, varia de acordo com o número de vetores gerados pelo comportamento do usuário na interação com a aplicação, e esse número também é influenciado pelo tamanho da senha, que pode varia entre seis e oito números.

Para cada um dos cenários, C1 a C3, o *framework* foi treinado em seis escopos diferentes:

a Escopo A(EA): utilizando todas as características capturadas nos Momentos 1 e 2, e gerando um modelo por usuário;

b Escopo B(EB): excluindo as características relacionadas aos dados de sensores, e gerando um modelo por usuário;

c Escopo C(EC): excluindo as características de tamanho do dedo e média do tamanho do dedo, e gerando um modelo por usuário;

d Escopo D(ED): utilizando todas as características capturadas nos Momentos 1 e 2, e gerando apenas um modelo para todos os usuários;

e Escopo E(EE): excluindo as características relacionadas aos dados de sensores, e gerando apenas um modelo para todos os usuários;

f Escopo F(EF): excluindo as características de tamanho do dedo e média do tamanho do dedo, e gerando apenas um modelo para todos os usuários.

O EA, é o mesmo utilizada em [41], já os outros foram criados visando encontrar a melhor performance do *framework* para os dados coletados e para a criação dos modelos nos cenários do experimento.

4.2.1 Resultados para Verificação Estática nos Escopos

Nesta seção será feita a análise dos resultados para verificação estática entre os escopos e cenários propostos. Nas tabelas o campo ALG indica o melhor algoritmo, o campo ALG(E), indica o algoritmo e escopo juntos, o campo QTD indica a quantidade de *templates* total para o usuário e o campo FAR_I indica o FAR do modelo em relação a tentativa de autenticação dos *templates* de todos os outros 50 usuários, impostores. As linhas das Tabelas em amarelo indicam que o passo 4 do *framework* foi atendido, o F1 de pelo menos 90% foi encontrado para o usuário, e as linhas em verde indicam que o modelo atendeu também o requisito do passo 5, ter um FAR menor que 10%, ou seja foi possível encontrar em um ou mais dos cenários um modelo que atendeu todos os requisitos do *framework*.

Para a verificação estática em EA para os cenários C1 a C3 foi possível encontrar um algoritmo com F1 a partir de 90%, para 80% dos usuários que ofereceram a quantidade mínima de *templates* para este cenário. A impossibilidade de criação de um modelo preciso para 20% dos usuários pode ter acontecido por ser um ambiente de teste, pois neste caso podemos ter vícios de uso, ou por que realmente não foi possível identificar um modelo para estes usuários entre os algoritmos e cenários testados. Os resultados dos melhores algoritmos por usuário para verificação estática em EA são detalhados nas Tabelas 4.4, 4.4 e 4.6.

Tabela 4.4: Detalhes dos melhores algoritmos VE Escopo A C1.

Usuário	**Melhores Algoritmos VE EA C1**								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
drds94uTXlk	NBG	39,39	0	19,69	100	71,73	83,54	80,30	12
cwdNwCqtFAs	NBG	0,54	0,81	0,67	99,18	99,45	99,32	99,32	11
eHC7qNMAdCI	GB	7,53	0	3,76	100	92,99	96,36	96,23	10
dUbEbDq40fM	NBB	0	10,71	5,35	89,28	100	94,33	94,64	12
fUB30EtiU0Q	GB	0	2,93	1,46	97,06	100	98,51	98,53	12
cWN_XjnNRDw	NBG	0	5,82	2,91	94,17	100	96,99	97,08	14

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
ffdzWINClJ4	NBB	0	9,74	4,87	90,25	100	94,87	95,12	14
dFOtPe4f8Xc	NBG	60,26	0	30,13	100	62,39	76,84	69,86	18
cBc3b9Cv4X0	NBG	0	2,45	1,22	97,54	100	98,75	98,77	17
ev9fChXnR3I	RF	4,29	40,67	22,48	59,32	93,24	72,51	77,51	17
fRA_pBm0ks4	NBB NBG	18,69	0	9,34	100	84,25	91,45	90,65	18
dyRTk2BUAeo	NBB	2,33	3,95	3,14	96,04	97,62	96,82	96,85	21
cFxPtyX-07w	SVM	3,63	0	1,81	100	96,48	98,21	98,18	18
dVGmimRO7bE	RF GB XGB	20,64	0	10,32	100	82,88	90,64	89,67	20
fGTU-LDm8uM	GB	6,82	32,42	19,62	67,57	90,82	77,49	80,37	20
fmVXDwdw20Q	NBB	0,20	8,47	4,34	91,52	99,77	95,47	95,65	25
eqmPzjjzrHk	NBG	13,80	0	6,90	100	87,87	93,54	93,10	23
enJGMKaiFiI	RF	8,61	5,53	7,07	94,46	91,63	93,03	92,92	27
fIewI06H8u8	XGB	3,09	0	1,54	100	96,99	98,47	98,45	31
eerZKZFi2kY	GB	38,66	18,37	28,50	81,62	67,87	74,11	71,49	40
dlj7Igoq3HQ	NBB	0,24	0,05	0,15	99,94	99,88	99,91	99,88	53
fDAVsmA3HUY	NBG	4,79	2,48	3,64	97,51	96,25	96,87	96,49	69
f0ttzpoZyeA	NBG	76,66	0	38,33	100	56,60	72,29	61,66	75
eoWIhgawcZ0	SVM	17,83	0,54	9,18	99,45	84,79	91,54	90,81	84
fhw9jzhvkGs	RF	53,62	1,24	27,43	98,75	64,08	78,25	72,56	90

Tabela 4.5: Detalhes dos melhores algoritmos VE Escopo A C2.

Usuário	Melhores Algoritmos VE EA C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	NBG	0	20,64	10,32	79,35	100	88,48	89,67	18
cBc3b9Cv4X0	NBG	0	4,79	2,39	95,20	100	97,54	97,60	17
ev9fChXnR3I	NBB	59,04	25,77	42,41	74,22	55,69	63,63	57,58	17
fRA_pBm0ks4	NBB NBG SVM GB XGB	29,85	0	14,92	100	77	87,01	85,07	18
dyRTk2BUAeo	NBG	7,59	12,62	10,12	87,35	91,99	89,61	89,87	21
cFxPtyX-07w	Todos	0	0	0	100	100	100	100	18
dVGmimRO7bE	NBG	0	1,84	0,92	98,15	100	99,06	99,07	20
fGTU-LDm8uM	XGB	22,05	2,25	12,15	97,74	81,58	88,93	87,84	20

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fmVXDwdw20Q	NBB	0	1,88	0,94	98,11	100	99,04	99,05	25
eqmPzjjzrHk	NBB	0	13	0,06	99,86	100	99,93	99,93	23
enJGMKaiFiI	XGB	9,74	0,79	5,26	99,20	91,05	94,95	94,73	27
fIewI06H8u8	RF	31,38	2,37	16,87	97,62	75,67	85,26	83,12	31
eerZKZFi2kY	GB	23,75	0	11,87	100	80,80	89,38	88,12	40
dlj7Igoq3HQ	NBB	0,66	0,57	0,61	99,42	99,66	99,54	99,39	53
fDAVsmA3HUY	NBG	4,75	2,71	3,73	97,28	96,00	96,64	96,34	69
f0ttzpoZyeA	NBG	13,26	0	6,63	100	88,28	93,77	93,36	75
eoWIhgawcZ0	NBB	18,82	0	9,41	100	84,15	91,39	90,58	84
fhw9jzhvkGs	GB	5,40	20,31	12,86	79,68	93,64	86,10	87,13	90

Tabela 4.6: Detalhes dos melhores algoritmos VE Escopo A C3.

Usuário	Melhores Algoritmos VE EA C3								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dyRTk2BUAeo	GB	10,48	0	5,24	100	90,51	95,01	94,75	21
dVGmimRO7bE	NBG	29,54	0	14,72	100	77,19	87,12	85,22	20
fGTU-LDm8uM	XGB	19,91	0	9,95	100	83,39	90,94	90,04	20
fmVXDwdw20Q	NBG	0	1,81	0,90	98,18	100	99,08	99,09	25
eqmPzjjzrHk	SVM	0,62	0	0,31	100	99,37	99,68	99,68	23
enJGMKaiFiI	GB	16,44	1,23	8,83	98,76	85,72	91,78	91,16	27
fIewI06H8u8	GB	1,92	30,36	16,14	69,63	97,30	81,17	83,85	31
eerZKZFi2kY	RF	58,53	0	29,26	100	63,07	77,35	70,73	40
dlj7Igoq3HQ	NBB	0,04	0,16	0,10	99,83	99,97	99,90	99,87	53
fDAVsmA3HUY	NBG	4,75	3,04	3,89	96,95	95,57	96,26	96,12	69
f0ttzpoZyeA	NBG	14,24	0	7,12	100	87,53	93,35	92,87	75
eoWIhgawcZ0	NBB	17,04	0	8,52	100	85,43	92,14	91,47	84
fhw9jzhvkGs	GB	13,73	15,93	14,83	84,06	85,95	85,00	85,16	90

Para a verificação estática em EB foi possível encontrar um algoritmo com F1 a partir de 90%, para 64% dos usuários, uma quantidade menor do que em EA, demonstrando a importância dos dados de sensores para a definição do modelo de verificação estática. Os resultados dos melhores algoritmos por usuário para verificação estática em EB são detalhados nas Tabelas 4.7, 4.8 e 4.9.

Tabela 4.7: Detalhes dos melhores algoritmos VE Escopo B C1.

Usuário	Melhores Algoritmos VE EB C1								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
drds94uTXlk	NBG	16,36	0	8,18	100	85,93	92,43	91,81	12
cwdNwCqtFAs	NBG	37,39	1,62	19,51	98,37	72,45	83,44	80,48	11

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
eHC7qNMAdCI	RF GB XGB	35,88	0	17,94	100	73,59	84,78	82,05	10
dUbEbDq40fM	NBG	0	3,57	1,78	96,42	100	98,18	98,21	12
fUB30EtiU0Q	NBB	29,61	6,35	17,99	93,64	75,95	83,88	82	12
cWN_XjnNRDw	NBG	13,24	6,99	10,11	93,00	87,53	90,19	89,88	14
ffdzWINClJ4	GB	41,18	6,64	23,91	93,35	69,39	79,60	76,08	14
dFOtPe4f8Xc	NBG	0	12,66	6,33	87,33	100	93,24	93,66	18
cBc3b9Cv4X0	NBG	0	4,06	2,03	95,93	100	97,92	97,96	17
ev9fChXnR3I	NBB	41,19	2,09	21,64	97,90	70,38	81,89	78,35	17
fRA_pBm0ks4	NBG	0	7,71	3,85	92,28	100	95,98	96,14	18
dyRTk2BUAeo	RF GB	30	0	15	100	76,92	86,95	85	21
cFxPtyX-07w	XGB	0	6,18	3,09	93,81	100	96,80	96,90	18
dVGmimRO7bE	NBG	0	8,44	4,22	91,55	100	95,58	95,77	20
fGTU-LDm8uM	NBG	59,17	9,66	34,42	90,33	60,42	72,41	65,57	20
fmVXDwdw20Q	NBG	0	5,78	2,89	94,21	100	97,01	97,10	25
eqmPzjjzrHk	Todos	0	0	0	100	100	100	100	23
enJGMKaiFiI	XGB	11,53	16,36	13,94	83,63	87,88	85,70	86,05	27
fIewI06H8u8	XGB	2,19	0	1,09	100	97,85	98,91	98,90	31
eerZKZFi2kY	NBG	39,79	0,28	20,04	99,71	71,47	83,26	79,95	40
dlj7Igoq3HQ	NBG	0	2,79	1,39	97,20	100	98,58	98,08	53
fDAVsmA3HUY	NBG	4,79	2,48	3,64	97,51	96,25	96,87	96,49	69
f0ttzpoZyeA	NBG	0	0,08	0,04	99,91	100	99,95	99,95	75
eoWIhgawcZ0	NBB	16,58	2,17	9,38	97,82	85,50	91,25	90,61	84
fhw9jzhvkGs	NBB SVM GB XGB	0	88	44	11,99	100	21,42	21,42	90

Tabela 4.8: Detalhes dos melhores algoritmos VE Escopo B C2.

Usuário	Melhores Algoritmos VE EB C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	NBG	0	20,64	10,32	79,35	100	88,48	89,67	18
cBc3b9Cv4X0	NBB	4,41	1,15	2,78	98,84	95,72	97,26	97,21	17
ev9fChXnR3I	NBB	37,29	0,36	18,82	99,63	72,76	84,10	81,17	17
fRA_pBm0ks4	NBB	0	3,54	1,77	96,45	100	98,19	98,22	18

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dyRTk2BUAeo	RF GB XGB	18,89	0	9,44	100	84,10	91,36	90,55	21
cFxPtyX-07w	NBG	53,32	0	26,66	100	65,22	78,95	73,33	18
dVGmimRO7bE	NBG	12,47	1,84	7,15	98,15	88,72	93,20	92,84	20
fGTU-LDm8uM	NBG	65,99	5,76	35,88	94,23	58,81	72,42	64,11	20
fmVXDwdw20Q	NBG	0	3,67	1,83	96,32	100	98,12	98,16	25
eqmPzjjzrHk	Todos	0	0	0	100	100	100	100	23
enJGMKaiFiI	GB	12,99	8,39	10,69	91,60	87,57	89,54	89,30	27
fIewI06H8u8	XGB	0	1,56	0,78	98,43	100	99,21	99,21	31
eerZKZFi2kY	NBG	42,85	1,03	21,94	98,96	69,78	81,85	78,05	40
dlj7Igoq3HQ	NBG	0	1,58	0,79	98,41	100	99,2	98,95	53
fDAVsmA3HUY	NBG	4,79	2,71	3,75	97,28	95,97	96,62	96,33	69
f0ttzpoZyeA	NBG	0	1,44	0,72	98,55	100	99,27	99,27	75
eoWIhgawcZ0	SVM RF	18,82	0	9,41	100	84,15	91,39	90,58	84
fhw9jzhvkGs	Todos	0	93,20	46,60	6,79	100	12,72	53,39	90

Tabela 4.9: Detalhes dos melhores algoritmos VE Escopo B C3.

Usuário	Melhores Algoritmos VE EB C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	RF GB XGB	37,76	0	18,84	100	72,58	84,11	81,11	21
dVGmimRO7bE	NBG	29,54	0	14,27	100	77,19	87,12	85,22	20
fGTU-LDm8uM	XGB	45,59	0	22,79	100	68,68	81,43	77,20	20
fmVXDwdw20Q	NBG	0	1,81	0,90	98,18	100	99,08	99,09	25
eqmPzjjzrHk	Todos	0	0	0	100	100	100	100	23
enJGMKaiFiI	XGB	12,97	15,10	14,03	84,89	86,74	82,81	85,96	27
fIewI06H8u8	SVM	0	1,86	0,93	98,13	100	99,05	99,06	31
eerZKZFi2kY	NBG	54,26	0,81	27,54	99,18	64,63	78,26	72,45	40
dlj7Igoq3HQ	NBG	0	1,60	0,8	98,39	100	99,19	98,98	53
fDAVsmA3HUY	NBG	4,75	3,08	3,91	96,91	95,57	96,24	96,10	69
f0ttzpoZyeA	NBG	0	1,65	0,82	98,34	100	99,16	99,17	75
eoWIhgawcZ0	SVM	20,11	0	10,05	100	83,25	90,86	89,94	84
fhw9jzhvkGs	Todos	0	98,8	49,4	1,2	100	2,37	50,6	90

Para a verificação estática em EC foi possível encontrar um algoritmo com F1 a partir de

90%, para 68% dos usuários. Demonstrando que as características ligadas ao tamanho do dedo quando removidas, tornam mais difícil a tarefa de encontrar um modelo com boa performance, reforçando a importância dessas características para a criação do modelo. Os resultados dos melhores algoritmos por usuário para verificação estática em EC são detalhados nas Tabelas 4.10, 4.11 e 4.12.

Tabela 4.10: Detalhes dos melhores algoritmos VE Escopo C C1.

Usuário	**Melhores Algoritmos VE EC C1**								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
drds94uTXlk	NBG	39,39	0	19,69	100	71,73	83,54	80,30	12
cwdNwCqtFAs	NBG	17,07	3,79	10,43	96,20	84,92	90,21	89,56	11
eHC7qNMAdCI	RF	0,25	0	0,12	100	99,74	99,87	99,87	10
dUbEbDq40fM	NBB	0	17,22	8,61	82,77	100	90,57	91,38	12
fUB30EtiU0Q	NBB	24,59	7,16	15,87	92,83	79,05	85,39	84,12	12
cWN_XjnNRDw	NBB	15,04	0,42	7,73	99,57	86,87	92,79	92,26	14
ffdzWINClJ4	NBB	0	9,74	4,87	90,25	100	94,87	95,12	14
dFOtPe4f8Xc	XGB	28,38	0	14,19	100	77,89	87,57	85,80	18
cBc3b9Cv4X0	NBG	0	3,4	1,52	96,95	100	98,45	98,47	17
ev9fChXnR3I	XGB	0	41,61	20,80	58,38	100	73,72	79,19	17
fRA_pBm0ks4	NBB NBG SVM GB XGB	18,69	0	9,34	100	84,25	91,45	90,65	18
dyRTk2BUAeo	NBB	2,28	7,05	4,67	92,94	97,6	95,21	95,32	21
cFxPtyX-07w	SVM	3	0	1,5	100	97,08	98,52	98,49	18
dVGmimRO7bE	XGB	8,08	20,2	14,14	79,79	90,80	84,94	85,85	20
fGTU-LDm8uM	XGB	41,54	0	20,77	100	70,64	82,8	79,22	20
fmVXDwdw20Q	NBB	0	11,44	5,72	88,55	100	93,93	94,27	25
eqmPzjjzrHk	NBG	13,8	0	6,9	100	87,87	93,54	93,1	23
enJGMKaiFiI	GB	7,45	11,88	9,66	88,11	92,19	90,11	90,33	27
fIewI06H8u8	XGB	22,1	10,32	16,21	89,67	80,22	84,68	83,78	31
eerZKZFi2kY	GB	22,72	30,39	26,55	69,6	75,39	72,38	73,44	40
dlj7Igoq3HQ	NBG	0,61	3,3	1,96	96,69	99,70	98,17	97,54	53
fDAVsmA3HUY	NBB	79,13	,24	40,18	98,75	61,17	75,54	64,32	69
f0ttzpoZyeA	NBG	76,66	0	38,33	100	56,6	72,29	61,66	75
eoWIhgawcZ0	SVM	11,2	1,14	6,17	98,85	89,82	94,12	93,82	84
fhw9jzhvkGs	XGB	50,29	13,36	31,82	86,63	63,26	73,13	68,17	90

Tabela 4.11: Detalhes dos melhores algoritmos VE Escopo C C2.

Usuário	Melhores Algoritmos VE EC C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	XGB	52,9	0	26,45	100	65,4	79,08	73,54	18
cBc3b9Cv4X0	GB	0	1,43	0,71	98,56	100	99,27	99,28	17
ev9fChXnR3I	NBB	58,86	26,14	42,5	73,85	55,64	63,47	57,49	17
fRA_pBm0ks4	GB	28,18	0	14,09	100	78,01	87,64	85,9	18
dyRTk2BUAeo	XGB	4,5	8	6,25	91,99	95,33	93,63	93,74	21
cFxPtyX-07w	Todos	0	0	0	100	100	100	100	18
dVGmimRO7bE	GB XGB	50,97	10,62	30,8	89,37	63,67	74,36	69,19	20
fGTU-LDm8uM	XGB	16,62	2,25	9,44	97,74	85,46	91,19	90,55	20
fmVXDwdw20Q	NBG	0	2,68	1,34	97,31	100	98,64	98,65	25
eqmPzjjzrHk	NBB	3,53	9,69	6,61	90,30	96,22	93,17	93,38	23
enJGMKaiFiI	XGB	9,19	0,79	4,99	99,2	91,52	95,2	95	27
fIewI06H8u8	NBB	50,9	2,37	26,63	97,62	65,72	78,56	73,36	31
eerZKZFi2kY	NBB	49,39	6,54	27,96	93,45	65,42	76,96	72,03	40
dlj7Igoq3HQ	NBG	0,61	11,07	5,84	88,92	99,64	93,97	92,46	53
fDAVsmA3HUY	NBB	31,77	22,85	27,31	77,14	74,02	75,55	73,04	69
f0ttzpoZyeA	NBG	13,26	0	6,63	100	88,28	93,77	93,36	75
eoWIhgawcZ0	NBB	19,86	0	9,93	100	83,42	90,96	90,06	84
fhw9jzhvkGs	XGB	53,45	0	26,72	100	65,16	78,9	73,27	90

Tabela 4.12: Detalhes dos melhores algoritmos VE Escopo C C3.

Usuário	Melhores Algoritmos VE EC C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	GB	3,62	0	1,81	100	96,49	98,21	98,18	21
dVGmimRO7bE	GB	51,03	0	25,51	100	66,21	79,67	74,48	20
fGTU-LDm8uM	RF	11,11	2,88	6,99	97,11	89,73	93,27	93	20
fmVXDwdw20Q	NBG	0	1,96	0,98	98,03	100	99	99,01	25
eqmPzjjzrHk	NBG RF GB XGB	28,87	0	14,43	100	77,59	87,38	85,56	23
enJGMKaiFiI	GB	16,55	2,79	9,67	97,2	85,44	90,94	90,32	27
fIewI06H8u8	GB	9,39	0	4,69	100	91,41	95,51	95,3	31
eerZKZFi2kY	RF	47,56	11,17	29,36	88,82	65,12	75,15	70,63	40
dlj7Igoq3HQ	NBG	0,61	1,38	1	98,61	99,63	99,12	98,89	53
fDAVsmA3HUY	GB	37,35	6,83	22,09	93,16	72,53	81,56	78,34	69

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
f0ttzpoZyeA	NBG	14,24	0	7,12	100	87,53	93,35	92,87	75
eoWIhgawcZ0	NBB	13,97	0	6,98	100	87,73	93,46	93,01	84
fhw9jzhvkGs	SVM	63,73	0	31,86	100	61,07	75,83	68,13	90

Para a verificação estática em ED foi possível encontrar um algoritmo com F1 a partir de 90%, para 68% dos usuários. E foi possível encontrar modelos com F1 desejado para aqueles usuários que não tiveram um modelo satisfatório em EA, um indicativo que esses dois escopos podem se complementar. Os resultados dos melhores algoritmos por usuário para verificação estática em ED são detalhados nas Tabelas 4.13, 4.14 e 4.15.

Tabela 4.13: Detalhes dos melhores algoritmos VE Escopo D C1.

Usuário	Melhores Algoritmos VE ED C1								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
drds94uTXlk	RF	0	10,90	5,45	89,09	100	94,23	99,94	12
cwdNwCqtFAs	RF	0,33	13,27	6,80	86,72	73,73	79,70	99,53	11
eHC7qNMAdCI	RF	0	0	0	100	100	100	100	10
dUbEbDq40fM	RF	0,61	0	0,30	100	69,38	81,92	99,39	12
fUB30EtiU0Q	RF	0,47	0	0,23	100	79,12	88,34	99,55	12
cWN_XjnNRDw	RF	0	0	0	100	100	100	100	14
ffdzWINClJ4	RF	0	7,97	3,98	92,02	100	95,84	99,68	14
dFOtPe4f8Xc	RF	0,75	9,60	5,18	90,39	44,32	59,48	99,18	18
cBc3b9Cv4X0	RF	3,30	23,85	13,57	76,14	56,68	64,99	95,58	17
ev9fChXnR3I	RF	0,07	45,17	22,62	54,82	95,43	69,64	98,68	17
fRA_pBm0ks4	RF	0	0	0	100	100	100	100	18
dyRTk2BUAeo	RF	0,14	0	0,07	100	97,71	98,84	99,86	21
cFxPtyX-07w	RF	0,04	0	0,02	100	98,56	99,27	99,95	18
dVGmimRO7bE	RF	0,2	18,29	9,24	81,70	94,31	87,55	99,09	20
fGTU-LDm8uM	RF	0,94	0	0,47	100	84,06	91,34	99,09	20
fmVXDwdw20Q	RF	0	19,22	9,61	80,77	100	89,36	99,19	25
eqmPzjjzrHk	RF	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF	0,14	1,51	0,82	98,48	97,23	97,85	99,78	27
fIewI06H8u8	RF	0	1,76	0,88	98,23	100	99,11	99,88	31
eerZKZFi2kY	RF	0,56	84,80	42,68	15,19	35,35	21,25	97,76	40
dlj7Igoq3HQ	RF	1,61	19,75	10,68	80,24	89,85	84,77	95,63	53
fDAVsmA3HUY	RF	5,81	6,02	5,91	93,77	60,90	73,90	94,16	69
f0ttzpoZyeA	RF	0,20	0	0,10	100	94,24	97,03	99,80	75
eoWIhgawcZ0	RF	0,09	0	0,04	100	98,34	99,16	99,91	84
fhw9jzhvkGs	RF	0	88,00	44,00	11,99	100	21,42	95,73	90

Tabela 4.14: Detalhes dos melhores algoritmos VE Escopo D C2.

Usuário	Melhores Algoritmos VE ED C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	RF	0,68	0	0,34	100	48,72	65,53	99,31	18
cBc3b9Cv4X0	RF	7,09	0	3,54	100	39,06	56,19	93,21	17
ev9fChXnR3I	RF	0,08	6,21	3,15	93,78	96,24	95	99,77	17
fRA_pBm0ks4	RF	0	0	0	0	100	100	100	18
dyRTk2BUAeo	RF	0,01	9,48	4,74	90,51	99,70	94,88	99,39	21
cFxPtyX-07w	RF	0	0	0	100	100	100	100	18
dVGmimRO7bE	RF	0	0,43	0,21	99,56	100	99,78	99,98	20
fGTU-LDm8uM	RF	0,33	1	0,67	98,99	93,89	93,89	96,38	20
fmVXDwdw20Q	RF	0	0	0	100	100	100	100	25
eqmPzjjzrHk	RF	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF	0,51	0,55	0,53	99,44	91,47	95,29	99,48	27
fIewI06H8u8	RF	0	1	0,5	98,99	100	99,49	99,91	31
eerZKZFi2kY	RF	0	47,48	23,92	52,15	100	68,55	98,84	40
dlj7Igoq3HQ	RF	0	34,39	17,19	65,60	100	79,22	93,21	53
fDAVsmA3HUY	RF	7,53	1,16	4,34	98,83	63,80	77,55	93,22	69
f0ttzpoZyeA	RF	0,16	0	0,08	100	96,47	98,20	99,84	75
eoWIhgawcZ0	RF	0	0	0	100	100	100	100	84
fhw9jzhvkGs	RF	0	93,20	46,60	6,79	100	12,72	93,81	90

Tabela 4.15: Detalhes dos melhores algoritmos VE Escopo D C3.

Usuário	Melhores Algoritmos VE ED C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	RF	0	0	0	100	100	100	100	21
dVGmimRO7bE	RF	0	0	0	100	100	100	100	20
fGTU-LDm8uM	RF	0,02	0	0,01	100	99,42	99,71	99,97	20
fmVXDwdw20Q	RF	2,11	0	1	100	65,83	79,39	97,97	25
eqmPzjjzrHk	RF	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF	0	0,44	0,22	99,55	100	99,77	99,97	27
fIewI06H8u8	RF	0	0	0	100	100	100	100	31
eerZKZFi2kY	RF	0,78	0	0,39	100	78,97	88,25	99,22	40
dlj7Igoq3HQ	RF	0	8,19	4,09	91,80	100	95,72	97,97	53
fDAVsmA3HUY	RF	9,13	0	4,56	100	66,16	79,63	92,24	69
f0ttzpoZyeA	RF	0,25	0	0,12	100	95,94	97,92	99,75	75
eoWIhgawcZ0	RF	0	0	0	100	100	100	100	84
fhw9jzhvkGs	RF	0	98,8	49,4	1,2	100	2,37	91,23	90

Para a verificação estática em EE foi possível encontrar um algoritmo com F1 a partir de 90%, para apenas 24% dos usuários, reforçando a importância dos dados de sensores para a definição do modelo de verificação estática, assim como aconteceu no EB. Os resultados dos melhores algoritmos por usuário para verificação estática em EE são detalhados nas Tabelas 4.16, 4.17 e 4.18.

Tabela 4.16: Detalhes dos melhores algoritmos VE Escopo E C1.

Usuário	**Melhores Algoritmos VE EE C1**								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
drds94uTXlk	XGB	0,02	0	0,01	100	95,93	97,92	99,97	12
cwdNwCqtFAs	XGB	0,37	0	0,18	100	74,09	85,12	99,62	11
eHC7qNMAdCI	XGB	1,02	35,37	18,20	64,62	59,25	61,82	98,20	10
dUbEbDq40fM	XGB	0	0	0	100	100	100	100	12
fUB30EtiU0Q	XGB	1,76	92,01	46,84	7,98	7,50	7,73	96,63	12
cWN_XjnNRDw	XGB	0	0	0	100	100	100	100	14
ffdzWINClJ4	XGB	1,37	70,33	35,85	29,66	46,74	36,29	95,93	14
dFOtPe4f8Xc	XGB	7,23	36,68	21,96	63,31	5,48	10,09	92,56	18
cBc3b9Cv4X0	XGB	4,33	40,53	22,43	59,46	43,83	50,46	93,71	17
ev9fChXnR3I	XGB	2,69	80,08	41,39	19,91	17,24	18,48	95,17	17
fRA_pBm0ks4	XGB	1,22	0	0,61	100	64,77	78,62	98,80	18
dyRTk2BUAeo	XGB	0,71	10,05	5,38	89,94	88,37	89,15	98,75	21
cFxPtyX-07w	XGB	3,92	31,30	17,61	68,69	36,36	47,55	95,20	18
dVGmimRO7bE	XGB	0	9,62	4,81	90,37	100	94,94	99,62	20
fGTU-LDm8uM	XGB	2,39	87,43	44,91	12,56	20,77	15,65	93,55	20
fmVXDwdw20Q	XGB	0,93	60,50	30,72	39,49	64,67	49,03	96,57	25
eqmPzjjzrHk	XGB	0	0	0	100	100	100	100	23
enJGMKaiFiI	XGB	2,62	59,17	30,90	40,82	44,67	42,66	94,57	27
fIewI06H8u8	XGB	0,11	0,64	0,38	99,35	98,38	98,86	99,84	31
eerZKZFi2kY	XGB	2,63	61,36	31,99	38,63	22,89	28,75	96,19	40
dlj7Igoq3HQ	XGB	2,47	28,77	15,62	71,22	83,68	76,95	93,53	53
fDAVsmA3HUY	XGB	2,01	67,29	34,65	32,70	61,02	42,58	92,24	69
f0ttzpoZyeA	XGB	1,09	32,31	16,70	67,68	67,08	67,38	97,89	75
eoWIhgawcZ0	XGB	2,29	26,59	14,44	73,40	64,13	68,45	96,42	84
fhw9jzhvkGs	XGB	0	88	44	11,99	100	21,42	95,73	90

Tabela 4.17: Detalhes dos melhores algoritmos VE Escopo E C2.

Usuário	Melhores Algoritmos VE EE C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	XGB	8,96	40	24,48	60	4,17	7,80	90,82	18
cBc3b9Cv4X0	XGB	9,74	30,68	20,21	69,31	24,45	36,15	89,34	17
ev9fChXnR3I	XGB	1,31	64,18	32,64	35,83	42,51	38,88	97,43	17
fRA_pBm0ks4	XGB	0,87	0	0,43	100	70	82,37	99,14	18
dyRTk2BUAeo	XGB	0	0	0	100	100	100	100	21
cFxPtyX-07w	XGB	2,33	46,84	24,59	53,15	36,95	43,59	96,54	18
dVGmimRO7bE	XGB	0	0	0	100	100	100	100	20
fGTU-LDm8uM	XGB	0,80	51,79	26,30	48,20	75,82	58,93	96,64	20
fmVXDwdw20Q	XGB	2,15	68,52	35,33	31,47	39,08	34,87	95,05	25
eqmPzjjzrHk	XGB	0	0	0	100	100	100	100	23
enJGMKaiFiI	XGB	2,47	23,85	13,16	76,14	63,14	69,03	96,40	27
fIewI06H8u8	XGB	0	0	0	100	100	100	100	31
eerZKZFi2kY	XGB	2,32	51,29	26,80	48,70	34,22	40,19	96,48	40
dlj7Igoq3HQ	XGB	0,99	49,57	25,28	50,42	92,58	65,28	89,41	53
fDAVsmA3HUY	XGB	3,38	49,92	26,65	50,07	66,51	57,13	91,09	69
f0ttzpoZyeA	XGB	1,16	38,84	20	61,15	70,35	65,43	97,19	75
eoWIhgawcZ0	XGB	2,41	17,68	10,04	82,31	72,75	77,24	96,47	84
fhw9jzhvkGs	XGB	0	93,20	46,60	6,79	100	12,72	93,81	90

Tabela 4.18: Detalhes dos melhores algoritmos VE Escopo E C3.

Usuário	Melhores Algoritmos VE EE C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	XGB	0	0	0	100	99,86	99,93	99,99	21
dVGmimRO7bE	XGB	0	0	0	100	100	100	100	20
fGTU-LDm8uM	XGB	0,87	6,78	3,82	93,21	82,08	87,29	98,88	20
fmVXDwdw20Q	XGB	6,47	29,19	17,83	70,80	30,78	42,91	92,63	25
eqmPzjjzrHk	XGB	0	0	0	100	100	100	100	23
enJGMKaiFiI	XGB	0,3	15,77	8,03	84,22	93,89	88,79	98,87	27
fIewI06H8u8	XGB	0	0,18	0,09	99,81	100	99,90	99,98	31
eerZKZFi2kY	XGB	5,19	53,65	29,42	46,34	21,09	28,98	93,39	40
dlj7Igoq3HQ	XGB	0	9,44	4,72	90,55	100	95,04	97,66	53
fDAVsmA3HUY	XGB	7,79	32,12	19,95	67,87	60,88	64,19	88,52	69
f0ttzpoZyeA	XGB	5,45	27,86	16,65	72,13	44,57	55,10	93,25	75
eoWIhgawcZ0	XGB	1,53	42,85	22,19	57,14	79,86	66,61	94,47	84
fhw9jzhvkGs	XGB	0	98,8	49,4	1,2	100	2,37	91,23	90

Para a verificação estática em EF foi possível encontrar um algoritmo com F1 a partir de 90%, para apenas 52% dos usuários, reforçando a importância dos dados de tamanho do dedo para a definição do modelo de verificação estática, assim como aconteceu no EC. Os resultados dos melhores algoritmos por usuário para verificação estática em EF são detalhados nas Tabelas 4.19, 4.20 e 4.21.

Tabela 4.19: Detalhes dos melhores algoritmos VE Escopo F C1.

Usuário	**Melhores Algoritmos VE EF C1**								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
drds94uTXlk	RF	0,52	0	0,26	100	47,55	64,43	99,47	12
cwdNwCqtFAs	RF	0,47	2,43	1,45	97,56	68,83	80,71	99,5	11
eHC7qNMAdCI	RF	0	0	0	100	100	100	100	10
dUbEbDq40fM	RF	0	0	0	100	100	100	100	12
fUB30EtiU0Q	RF	0,11	0	0,057	100	94,02	96,92	99,87	12
cWN_XjnNRDw	RF	0,55	45,02	22,78	54,97	73,61	62,94	98,24	14
ffdzWINClJ4	RF	0,01	5,46	2,73	94,53	99,61	97	99,77	14
dFOtPe4f8Xc	RF	0,83	13,53	7,18	86,46	40,65	55,3	99,07	18
cBc3b9Cv4X0	RF	7,62	14,65	11,13	85,34	38,9	3,44	92	17
ev9fChXnR3I	RF	0,86	65,19	33,03	34,8	53,2	42,07	97,36	17
fRA_pBm0ks4	RF	1,25	0	0,62	100	64,28	78,26	98,77	18
dyRTk2BUAeo	RF	1,46	50,2	25,83	49,79	67,09	57,16	95,76	21
cFxPtyX-07w	RF	0	0	0	100	100	100	100	18
dVGmimRO7bE	RF	3,09	61,71	32,4	38,28	33,54	33,75	94,61	20
fGTU-LDm8uM	RF	2,79	16,42	9,6	83,57	59,96	69,82	96,55	20
fmVXDwdw20Q	RF	0,32	54,3	27,31	45,69	86,1	59,7	97,42	25
eqmPzjjzrHk	RF	0,05	0	0,02	100	98,23	99,1	99,94	23
enJGMKaiFiI	RF	2,84	2,67	2,76	97,32	63,99	77,21	97,16	27
fIewI06H8u8	RF	0,48	63,35	31,91	36,64	84,44	51,1	95,3	31
eerZKZFi2kY	RF	0,81	40,23	20,52	59,76	59,85	59,81	98,4	40
dlj7Igoq3HQ	RF	1,23	36,69	18,96	63,3	90,17	74,38	93,39	53
fDAVsmA3HUY	RF	4,33	51,81	28,07	48,18	51,72	49,88	91,49	69
f0ttzpoZyeA	RF	1,63	0,08	0,88	99,91	66,4	79,78	98,37	75
eoWIhgawcZ0	RF	0	0	0	100	100	100	100	84
fhw9jzhvkGs	RF	0,1	42,81	21,46	57,18	96,49	71,81	97,82	90

Tabela 4.20: Detalhes dos melhores algoritmos VE Escopo F C2.

Usuário	Melhores Algoritmos VE EF C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	XGB	1,66	0	0,83	100	28,13	43,9	98,34	18
cBc3b9Cv4X0	XGB	6,57	0	3,28	100	40,88	58,04	93,7	17
ev9fChXnR3I	XGB	4,22	48,26	26,24	51,73	22,24	31,11	94,77	17
fRA_pBm0ks4	XGB	0,03	0	0,01	100	98,15	99,06	99,96	18
dyRTk2BUAeo	XGB	0,8	66,3	33,55	33,69	73,35	46,17	95,12	21
cFxPtyX-07w	XGB	0	0	0	100	100	100	100	18
dVGmimRO7bE	XGB	3,24	44,79	24,01	55,2	40,52	46,74	95,16	20
fGTU-LDm8uM	XGB	0,68	22,05	11,37	77,94	85,67	81,62	98,24	20
fmVXDwdw20Q	XGB	0,43	0,39	0,41	99,6	91,01	95,11	99,57	25
eqmPzjjzrHk	XGB	0	9,3	4,65	90,69	100	95,12	99,7	23
enJGMKaiFiI	XGB	1,84	0,23	1,03	99,76	75,07	85,67	98,24	27
fIewI06H8u8	XGB	5,44	38,24	21,84	61,75	50,55	55,59	91,84	31
eerZKZFi2kY	XGB	0,01	47,84	23,93	52,15	98,69	68,24	98,82	40
dlj7Igoq3HQ	XGB	0,36	36,08	18,22	63,91	97,7	77,27	92,58	53
fDAVsmA3HUY	XGB	2,45	37,46	19,95	62,53	77,41	69,18	93,39	69
f0ttzpoZyeA	XGB	0,03	1,34	0,69	98,65	99,22	98,93	99,9	75
eoWIhgawcZ0	XGB	0	0	0	100	100	100	100	84
fhw9jzhvkGs	XGB	0,21	33,01	16,61	66,98	95,6	78,77	97,6	90

Tabela 4.21: Detalhes dos melhores algoritmos VE Escopo F C3.

Usuário	Melhores Algoritmos VE EF C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	XGB	0,61	0	0,3	100	88,25	93,76	99,41	21
dVGmimRO7bE	XGB	4,81	45,24	25,03	54,75	25,11	34,43	94,02	20
fGTU-LDm8uM	XGB	3,53	0,14	1,84	99,85	54,7	70,68	96,6	20
fmVXDwdw20Q	XGB	0,05	0	0,02	100	98,65	99,32	99,94	25
eqmPzjjzrHk	XGB	0,0	2,51	1,25	97,48	100	98,72	99,92	23
enJGMKaiFiI	XGB	0,82	10,73	5,78	89,26	85,8	87,49	98,65	27
fIewI06H8u8	XGB	0,36	47,35	23,83	52,64	94,52	67,62	95,2	31
eerZKZFi2kY	XGB	1,03	3,86	2,44	96,13	73,56	83,34	98,88	40
dlj7Igoq3HQ	XGB	0	1,17	0,58	98,82	100	99,41	99,71	53
fDAVsmA3HUY	XGB	2,17	40,35	21,26	59,64	83,04	69,42	92,03	69
f0ttzpoZyeA	XGB	0	1,13	0,56	98,86	100	99,42	99,93	75
eoWIhgawcZ0	XGB	0	0	0	100	100	100	100	84
fhw9jzhvkGs	XGB	0,61	1,79	1,2	98,2	93,94	96,02	99,27	90

Diante dos resultados capturados nos escopos de verificação estática de EA a EF, foi definido que o *framework* incorporará os escopos ED, EA e EB, nesta ordem de precedência, pois apesar de EA apresentar uma melhor performance em relação ao ED, em ED o modelo já foi testado com dados de todos os outros usuários que geraram o modelo, além desta abordagem propiciar a diminuição de possíveis *overfittings* que possam acontecer em modelos individuais que só foram treinados com duas classes, os usuários legítimos(1) e os impostores(0), já que em um modelo compartilhado as classes internas podem ter uma acurácia de até 100% sem haver necessariamente o *overfitting* do modelo.

Os escopos EA e EB somente serão utilizados caso não seja possível encontrar o F1 de pelo menos 90% para o usuário em ED. Caso não seja encontrado F1 de pelo menos 90% para a classe em ED, será então efetuado o treinamento em EA, se ainda assim, não for possível encontrar o F1 definido será efetuado o treinamento em EB. Não sendo possível encontrar em nenhum dos cenários pode ser um indicativo de que o padrão de uso do usuário não pode ser capturado com o sistema e cenários propostos. O resultado deste experimento para o passo 4, encontrar o F1 de pelo menos 90%, do *framework* proposto com a união dos escopos EA a ED é demonstrado nas Tabelas 4.22, e4.23 e 4.24.

Tabela 4.22: Detalhes dos melhores algoritmos candidatos VE *Framework* C1.

Usuário	**Melhores Algoritmos Candidatos VE *Framework* C1**								
Identificação	**ALG (E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
drds94uTXlk	RF (ED)	0	10,90	5,45	89,09	100	94,23	99,94	12
cwdNwCqtFAs	NBG (EA)	0,54	0,81	0,67	99,18	99,45	99,32	99,32	11
eHC7qNMAdCI	RF (ED)	0	0	0	100	100	100	100	10
dUbEbDq40fM	NBB (EA)	0	10,71	5,35	89,28	100	94,33	94,64	12
fUB30EtiU0Q	GB (EA)	0	2,93	1,46	97,06	100	98,51	98,53	12
cWN_XjnNRDw	RF (ED)	0	0	0	100	100	100	100	14
ffdzWINClJ4	RF (ED)	0	7,97	3,98	92,02	100	95,84	99,68	14
dFOtPe4f8Xc	NBG (EB)	0	12,66	6,33	87,33	100	93,24	93,66	18
cBc3b9Cv4X0	NBG (EA)	0	2,45	1,22	97,54	100	98,75	98,77	17

Identificação	ALG (E)	FAR	FRR	EER	REC	PRC	F1	AC	QTD
ev9fChXnR3I	NBB (EB)	41,19	2,09	21,64	97,9	70,38	81,89	78,35	17
fRA_pBm0ks4	RF (ED)	0	0	0	100	100	100	100	18
dyRTk2BUAeo	RF (ED)	0,14	0	0,07	100	97,71	98,84	99,86	21
cFxPtyX-07w	RF (ED)	0,04	0	0,02	100	98,56	99,27	99,95	18
dVGmimRO7bE	RF GB XGB (EA)	20,64	0	10,32	100	82,88	90,64	89,67	20
fGTU-LDm8uM	RF (ED)	0,94	0	0,47	100	84,06	91,34	99,09	20
fmVXDwdw20Q	NBB (EA)	0,2	8,47	4,34	91,52	99,77	95,47	95,65	25
eqmPzjjzrHk	RF (ED)	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF (ED)	0,14	1,51	0,82	98,48	97,23	97,85	99,78	27
fIewI06H8u8	RF (ED)	0	1,76	0,88	98,23	100	99,11	99,88	31
eerZKZFi2kY	NBG (EB)	39,79	0,28	20,04	99,71	71,47	83,26	79,95	40
dlj7Igoq3HQ	NBB (EA)	0,24	0,05	0,15	99,94	99,88	99,91	99,88	53
fDAVsmA3HUY	NBG (EA)	4,79	2,48	3,64	97,51	96,25	96,87	96,49	69
f0ttzpoZyeA	RF (ED)	0,20	0	0,10	100	94,24	97,03	99,80	75
eoWIhgawcZ0	RF (ED)	0,09	0	0,04	100	98,34	99,16	99,91	84
fhw9jzhvkGs	RF (EA)	53,62	1,24	27,43	98,75	64,08	78,25	72,56	90

Tabela 4.23: Detalhes dos melhores algoritmos candidatos VE *Framework* C2.

Usuário	**Melhores Algoritmos Cenário VE *Framework* C2**								
Identificação	**ALG (E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	NBG (EA)	0	20,64	10,32	79,35	100	88,48	89,67	18
cBc3b9Cv4X0	NBG (EA)	0	4,79	2,39	95,20	100	97,54	97,60	17
ev9fChXnR3I	RF (ED)	0,08	6,21	3,15	93,78	96,24	95	99,77	17
fRA_pBm0ks4	RF (ED)	0	0	0	0	100	100	100	18
dyRTk2BUAeo	RF (ED)	0,01	9,48	4,74	90,51	99,70	94,88	99,39	21
cFxPtyX-07w	RF (ED)	0	0	0	100	100	100	100	18
dVGmimRO7bE	RF (ED)	0	0,43	0,21	99,56	100	99,78	99,98	20
fGTU-LDm8uM	RF (ED)	0,33	1	0,67	98,99	93,89	93,89	96,38	20
fmVXDwdw20Q	RF (ED)	0	0	0	100	100	100	100	25
eqmPzjjzrHk	RF (ED)	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF (ED)	0,51	0,55	0,53	99,44	91,47	95,29	99,48	27
fIewI06H8u8	RF (ED)	0	1	0,5	98,99	100	99,49	99,91	31
eerZKZFi2kY	GB (EA)	23,75	0	11,87	100	80,8	89,38	88,12	40
dlj7Igoq3HQ	NBB (EA)	0,66	0,57	0,61	99,42	99,66	99,54	99,39	53
fDAVsmA3HUY	NBG (EA)	4,75	2,71	3,73	97,28	96	96,64	96,34	69
f0ttzpoZyeA	RF (ED)	0,16	0	0,08	100	96,47	98,20	99,84	75
eoWIhgawcZ0	RF (ED)	0	0	0	100	100	100	100	84

Identificação	ALG (E)	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fhw9jzhvkGs	GB (EA)	5,4	20,31	12,86	79,68	93,64	86,1	87,13	90

Tabela 4.24: Detalhes dos melhores algoritmos candidatos VE *Framework* C3.

Usuário	Melhores Algoritmos Candidatos VE *Framework* C3								
Identificação	**ALG (E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dyRTk2BUAeo	RF (ED)	0	0	0	100	100	100	100	21
dVGmimRO7bE	RF (ED)	0	0	0	100	100	100	100	20
fGTU-LDm8uM	RF (ED)	0,02	0	0,01	100	99,42	99,71	99,97	20
fmVXDwdw20Q	NBG (EA)	0	1,81	0,9	98,18	100	99,08	99,09	25
eqmPzjjzrHk	RF (ED)	0	0	0	100	100	100	100	23
enJGMKaiFiI	RF (ED)	0	0,44	0,22	99,55	100	99,77	99,97	27
fIewI06H8u8	RF (ED)	0	0	0	100	100	100	100	31
eerZKZFi2kY	RF (ED)	0,78	0	0,39	100	78,97	88,25	99,22	40
dlj7Igoq3HQ	RF (ED)	0	8,19	4,09	91,80	100	95,72	97,97	53
fDAVsmA3HUY	NBG (EA)	4,75	3,04	3,89	96,95	95,57	96,26	96,12	69
f0ttzpoZyeA	RF (ED)	0,25	0	0,12	100	95,94	97,92	99,75	75
eoWIhgawcZ0	RF (ED)	0	0	0	100	100	100	100	84
fhw9jzhvkGs	GB (EA)	13,73	15,93	14,83	84,06	85,95	85	85,16	90

Conforme resultado detalhado na Tabela 4.22 para o cenário 1, foi possível encontrar um algoritmo com F1 a partir de 90% para 88% dos usuários. Para o cenário 2 para 83,3% dos usuários, conforme resultado detalhado na Tabela 4.23. No cenário 3 para 84,61% dos usuários,

conforme resultado detalhado na Tabela 4.24. Analisando todos os cenários para VE apenas não foi possível encontrar para dois dos 25 usuários.

Com a abordagem proposta, unindo os escopos EA, EB e ED no *framework* foi possível encontrar em um ou mais cenários, um algoritmo com a F1 pretendido para 92% dos usuários, não foi encontrado apenas para dois usuários, tendo sido o Escopo D responsável por solucionar a busca para 52% dos usuários em C1, 66% em C2 e 62,23%, o Escopo A para 32% dos usuários em C1, para 16,66% em C2 e 15,38% em C3 e o Escopo B para 4% dos usuários em C1, para 0% em C2 e 0% em C3, indicando assim que o escopo D é bem abrangente para encontrar o melhor algoritmo diante dos dados coletados e analisados neste experimento para a verificação estática.

O resultado para o passo 5 do *framework* proposto, modelos com FAR de até 10% para os *templates* dos outros usuários, são detalhados na Tabela 4.25. Este é o último crivo que o modelo deve passar para ser considerado apto para a autenticação do usuário.

Tabela 4.25: Resultados finais algoritmos por cenário VE *Framework*.

Usuário	**VE *Framework* Resultado Final**								
Identificação	**ALG (E) C1**	**F1 C1**	**FAR_I C1**	**ALG (E) C2**	**F1 C2**	**FAR_I C2**	**ALG (E) C3**	**F1 C3**	**FAR_I C3**
drds94uTXlk	RF (ED)	94,23	0	—	—	—	—	—	—
cwdNwCqtFAs	NBG (EA)	99,32	0	—	—	—	—	—	—
eHC7qNMAdCI	RF (ED)	100	0,6	—	—	—	—	—	—
dUbEbDq40fM	NBB (EA)	94,33	6,08	—	—	—	—	—	—
fUB30EtiU0Q	GB (EA)	98,51	44,41	—	—	—	—	—	—
cWN_XjnNRDw	RF (ED)	100	13,93	—	—	—	—	—	—
ffdzWINClJ4	RF (ED)	95,84	0	—	—	—	—	—	—
dFOtPe4f8Xc	NBG (EB)	93,24	11,15	—	—	—	—	—	—
cBc3b9Cv4X0	NBG (EA)	98,75	4,1	NBG (EA)	97,54	0	—	—	—
ev9fChXnR3I	—	—	—	RF (ED)	95	1,08	—	—	—
fRA_pBm0ks4	RF (ED)	100	1,51	RF (ED)	100	3,72	—	—	—

Identificação	ALG (E) C1	F1 C1	FAR_I C1	ALG (E) C2	F1 C2	FAR_I C2	ALG (E) C3	F1 C3	FAR_I C3
dyRTk2BUAeo	RF (ED)	98,84	0,14	RF (ED)	94,88	0,37	RF (ED)	100	0,03
cFxPtyX-07w	RF (ED)	99,27	0,09	RF (ED)	100	0,4	—	—	—
dVGmimRO7bE	RF GB XGB (EA)	90,64	77,75	RF (ED)	99,78	0,06	RF (ED)	100	0,03
fGTU-LDm8uM	RF (ED)	91,34	0,09	RF (ED)	93,89	0,05	RF (ED)	99,71	0,03
fmVXDwdw20Q	NBB (EA)	95,47	6,66	RF (ED)	100	0,01	RF (ED)	99,08	0,02
eqmPzjjzrHk	RF (ED)	100	1,62	RF (ED)	100	1,69	RF (ED)	100	1,42
enJGMKaiFiI	RF (ED)	97,85	0,2	RF (ED)	95,29	0,29	RF (ED)	99,77	0,64
fIewI06H8u8	RF (ED)	99,11	2,35	RF (ED)	99,49	0,29	RF (ED)	100	1,32
eerZKZFi2kY	—	—	—	—	—	—	—	—	—
dlj7Igoq3HQ	NBB (EA)	99,91	9,06	NBB (EA)	99,54	10,35	RF (ED)	95,72	0,58
fDAVsmA3HUY	NBG (EA)	96,87	0	NBG (EA)	96,64	8,69	NBG (EA)	96,26	8,85
f0ttzpoZyeA	RF (ED)	97,03	0,78	RF (ED)	98,2	1,9	RF (ED)	97,92	2,05
eoWIhgawcZ0	RF (ED)	99,16	2,29	RF (ED)	100	0,62	RF (ED)	100	2,14
fhw9jzhvkGs	—	—	—	—	—	—	—	—	—

Diante dos resultados demonstrados na Tabela 4.25 para o passo final do *framework* proposto, foi possível encontrar um modelo com F1 de pelo menos 90% e com FAR_I de até 10% para 80% dos usuários em VE, sendo que dos 5 usuários, 3 só ofereceram amostras suficientes para participar de C1. Indicando que a utilização do *framework* proposto para VE, se utilizado em conjunto com métodos convencionais como a senha, pode oferecer uma linha adicional de segurança com bom desempenho. No caso deste experimento em um ambiente de teste, com dispositivos distintos, em escopos variados conseguiu encontrar um modelo de qualidade para 20 dos 25 usuários, a maioria dos usuários que participaram do experimento.

4.2.2 Resultados para Verificação Dinâmica nos Escopos

Nesta seção será feita a análise dos resultados para verificação dinâmica entre os escopos e cenários propostos. Na tabelas o campo ALG indica o melhor algoritmo, o campo ALG(E), indica o algoritmo e escopo juntos, o campo QTD indica a quantidade de *templates* total para o usuário e o campo FAR_I indica o FAR do modelo em relação a tentativa de autenticação dos *templates* de todos os outros 50 usuários. As linhas das tabelas em amarelo indicam que o passo 4 do *framework* foi atendido, o F1 de pelo menos 90% foi encontrado para o usuário, e as linhas em verde indicam que o modelo atendeu também o requisito do passo 5, ter um FAR de no máximo 10%.

Para a verificação dinâmica em EA foi possível encontrar um algoritmo com F1 de pelo menos 90%, para 56,52% dos usuários. Diferente da verificação estática, que contém apenas uma tela para definição do modelo, na verificação dinâmica o padrão é formado pela captura das características do padrão de interação com várias telas. Isso pode ter influenciado a questão da porcentagem de 43,47% de usuários para os quais não foi possível encontrar um F1 a partir de 90%. Os resultados dos melhores algoritmos por usuário para verificação dinâmica em EA são detalhados nas Tabelas 4.26, 4.27 e 4.28.

Tabela 4.26: Detalhes dos melhores algoritmos VD Escopo A C1.

Usuário	**Melhores Algoritmos VD EA C1**								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	SVM	60,09	31,92	46,00	68,07	53,11	59,67	53,99	30
dUbEbDq40fM	NBB	45,37	0	22,68	100	68,78	81,50	77,31	36
fUB30EtiU0Q	XGB	0,99	0	0,49	100	99,01	99,50	99,50	36
cWN_XjnNRDw	NBB	27,19	1,03	14,11	98,96	78,44	87,52	85,88	42
ffdzWINClJ4	NBB	33,20	0	16,60	100	75,07	85,76	83,40	46
dFOtPe4f8Xc	XGB	8,30	16,02	12,16	83,97	90,99	87,34	87,83	65
cBc3b9Cv4X0	GB	0,31	42,47	21,39	57,52	99,45	72,88	78,60	51
ev9fChXnR3I	NBB	50,11	41,59	45,85	58,40	53,82	56,02	54,14	48
fRA_pBm0ks4	NBB	35,58	0	17,79	100	73,75	84,89	82,20	54
dyRTk2BUAeo	GB	32,54	5,38	18,96	94,61	74,40	83,30	81,03	53
cFxPtyX-07w	NBG	0	5,50	2,75	94,49	100	97,16	97,24	57
dVGmimRO7bE	GB	56,16	0	28,08	100	64,03	78,07	71,91	57
fGTU-LDm8uM	NBG	59,78	0	29,89	100	62,58	76,98	70,10	60
fmVXDwdw20Q	SVM	0,49	19,01	9,75	80,98	99,39	89,24	90,24	63
eqmPzjjzrHk	NBB	0	0,14	0,07	99,85	100	99,92	99,92	73
enJGMKaiFiI	XGB	27,01	0,86	13,94	99,13	78,58	87,67	86,05	80
fIewI06H8u8	NBG	66,44	0	33,22	100	60,08	78,06	66,77	90
eerZKZFi2kY	XGB	,77	49,80	25,28	50,19	98,48	66,49	74,71	120
dlj7Igoq3HQ	GB	39,49	0	19,74	100	64,03	78,07	71,91	161

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fDAVsmA3HUY	NBG	0	46,53	23,26	53,46	100	69,68	76,73	210
f0ttzpoZyeA	NBG	18,53	0	9,26	100	84,36	91,51	90,73	225
eoWIhgawcZ0	NBB	3,20	13,17	8,19	86,82	97,21	91,72	91,17	253
fhw9jzhvkGs	NBG RF	61,68	0	30,84	100	61,85	76,42	69,15	270

Tabela 4.27: Detalhes dos melhores algoritmos VD Escopo A C2.

Usuário	Melhores Algoritmos VD EA C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	GB	17,91	17,34	17,63	82,65	82,18	82,42	82,36	65
cBc3b9Cv4X0	XGB	48,75	0	24,37	100	67,22	80,39	75,62	51
ev9fChXnR3I	NBB RF XGB	100	0	50	100	50	66,66	50	48
fRA_pBm0ks4	SVM	30,24	0,35	15,30	99,64	76,71	86,68	84,69	54
dyRTk2BUAeo	SVM	0,28	0	0,14	100	99,71	99,85	99,85	53
cFxPtyX-07w	NBG	0	13,78	6,89	86,21	100	92,59	93,10	57
dVGmimRO7bE	NBG	96,32	0	48,16	100	50,93	67,49	51,83	57
fGTU-LDm8uM	XGB	44,54	0	22,27	100	69,18	81,78	77,72	60
fmVXDwdw20Q	SVM	2,07	0	1,03	100	97,96	98,97	98,96	63
eqmPzjjzrHk	SVM	1,29	0	0,64	100	98,72	99,35	99,35	73
enJGMKaiFiI	XGB	21,21	0	10,60	100	82,49	90,40	89,39	80
fIewI06H8u8	NBG	72	0	36	100	58,13	73,52	63,99	90
eerZKZFi2kY	GB XGB	31,78	0	15,89	100	75,88	86,28	84,10	120
dlj7Igoq3HQ	GB	16,71	0	8,35	100	85,67	92,28	91,64	161
fDAVsmA3HUY	NBG	0	5,44	2,72	94,55	100	97,19	97,27	210
f0ttzpoZyeA	NBG RF GB	21,64	0	10,82	100	82,20	90,23	89,17	225
eoWIhgawcZ0	NBB	3,70	6,91	5,31	93,08	96,78	94,89	94,54	253
fhw9jzhvkGs	RF	3,50	13,93	8,72	86,06	96,08	90,79	91,27	270

Tabela 4.28: Detalhes dos melhores algoritmos VD Escopo A C3.

Usuário	Melhores Algoritmos VD EA C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG	95,00	0	47,50	100	51,28	67,79	52,50	60
fmVXDwdw20Q	SVM	1,04	0	0,52	100	98,96	99,48	99,47	63
eqmPzjjzrHk	RF	55,45	0	27,72	100	64,32	78,29	72,27	73
enJGMKaiFiI	XGB	0,71	0	0,35	100	99,29	99,64	99,64	80
fIewI06H8u8	SVM	62	0	31	100	61,72	76,33	68,99	90
eerZKZFi2kY	XGB	33,44	0	16,72	100	74,93	85,67	83,27	120
dlj7Igoq3HQ	SVM	16,74	0,19	8,46	99,80	85,63	92,17	91,53	161
fDAVsmA3HUY	RF GB XGB	27,28	0	13,64	100	78,56	87,99	86,35	210
f0ttzpoZyeA	NBG	24,80	0	12,40	100	80,12	88,96	87,59	225
eoWIhgawcZ0	NBB	12,98	2,07	7,52	97,92	89,23	93,38	92,73	253
fhw9jzhvkGs	GB	1,11	0,60	0,86	99,39	98,88	99,14	99,13	270

Para a verificação dinâmica em EB foi possível encontrar um algoritmo com F1 a partir de 90%, para 60,86% dos usuários. Um indicativo de que os dados de sensores possam não ter a mesma influência que teve na criação dos modelos de verificação estática. Os resultados dos melhores algoritmos por usuário para verificação dinâmica são detalhados nas Tabelas 4.29, 4.30 e 4.31.

Tabela 4.29: Detalhes dos melhores algoritmos VD Escopo B C1.

Usuário	Melhores Algoritmos VD EB C1								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	NBG	0	6,57	3,28	93,42	100	96,6	96,71	30
dUbEbDq40fM	NBG	0	2,8	1,4	97,19	100	98,57	98,59	36
fUB30EtiU0Q	XGB	28,21	0	14,1	100	77,99	87,63	85,89	36
cWN_XjnNRDw	NBG	23,32	0	11,66	100	81,08	89,55	88,33	42
ffdzWINClJ4	NBG	32,4	0	16,2	100	75,52	86,05	83,8	46
dFOtPe4f8Xc	NBG	0	1,78	0,89	98,21	100	99,1	99,1	65
cBc3b9Cv4X0	XGB	0	0,15	0,07	99,84	100	99,92	99,92	51
ev9fChXnR3I	NBG	93,60	0,11	46,86	99,88	51,62	68,06	53,13	48
fRA_pBm0ks4	NBG	0	0,67	0,33	99,32	100	99,66	99,66	54
dyRTk2BUAeo	NBG	92,78	0	46,39	100	51,87	68,31	53,60	53
cFxPtyX-07w	NBG	47,79	5,55	26,67	94,44	66,39	77,97	73,32	57
dVGmimRO7bE	NBG	17,69	0	8,84	100	84,96	91,87	91,15	57
fGTU-LDm8uM	NBG	55,97	0,39	28,18	99,6	64,02	77,94	71,81	60

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fmVXDwdw20Q	NBG	0	8,83	4,41	91,16	100	95,37	95,58	63
eqmPzjjzrHk	NBB	0	2,2	1,1	97,79	100	98,88	98,89	73
enJGMKaiFiI	NBG	37,17	5,7	21,43	94,29	71,72	81,47	78,56	80
fIewI06H8u8	NBG	37,37	1,61	19,49	98,38	72,47	83,46	80,50	90
eerZKZFi2kY	NBG	0	71,42	35,71	28,57	100	44,44	64,28	120
dlj7Igoq3HQ	NBG	0	3,34	1,67	96,65	100	98,29	98,32	161
fDAVsmA3HUY	NBG	0	0,11	0,05	99,88	100	99,94	99,94	210
f0ttzpoZyeA	NBg	0	23,45	11,72	76,54	100	86,71	88,27	225
eoWIhgawcZ0	SVM	20,94	4,3	12,62	95,69	85,47	90,29	88,42	253
fhw9jzhvkGs	SVM	11,27	21,07	16,17	78,92	87,5	82,99	83,82	270

Tabela 4.30: Detalhes dos melhores algoritmos VD Escopo B C2.

Usuário	Melhores Algoritmos VD EB C2								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dFOtPe4f8Xc	NBG	0	3,46	1,73	96,53	100	98,23	98,26	65
cBc3b9Cv4X0	NBG	0	0,31	0,15	99,68	100	99,84	99,84	51
ev9fChXnR3I	NBG	92,36	0	46,18	100	51,98	68,40	53,81	48
fRA_pBm0ks4	SVM	48,75	0	24,37	100	67,22	80,4	75,62	54
dyRTk2BUAeo	NBB	92,3	0	46,15	100	52	68,42	53,84	53
cFxPtyX-07w	NBG	46,49	9,89	28,19	90,1	65,96	76,16	71,8	57
dVGmimRO7bE	NBG	21,63	0	10,81	100	82,21	90,23	89,18	57
fGTU-LDm8uM	NBG	82,59	0,46	41,53	99,53	54,64	70,55	58,46	60
fmVXDwdw20Q	NBG	0	0,83	0,41	99,16	100	99,58	99,58	63
eqmPzjjzrHk	Todos	0	0	0	100	100	100	100	73
enJGMKaiFiI	XGB	24	0	12	100	80,64	89,28	88	80
fIewI06H8u8	GB	21,95	,80	11,87	98,19	81,72	89,21	88,12	90
eerZKZFi2kY	RF XGB	31,78	0	15,89	100	75,88	86,28	84,1	120
dlj7Igoq3HQ	NBG	0	5,73	2,86	94,26	100	97,04	97,13	161
fDAVsmA3HUY	NBG	0	0,12	0,06	99,87	100	99,93	99,93	210
f0ttzpoZyeA	NBG	0	0,22	0,11	99,77	100	99,88	99,88	225
eoWIhgawcZ0	NBG	0	11,61	5,8	88,38	100	93,83	93,66	253
fhw9jzhvkGs	SVM	28,54	5,84	17,19	94,15	76,73	84,55	82,8	270

Tabela 4.31: Detalhes dos melhores algoritmos VD Escopo B C3.

Usuário	Melhores Algoritmos VD EB C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG	92,5	0	46,25	100	51,94	68,37	53,75	60
fmVXDwdw20Q	XGB	2,43	0	1,21	100	97,61	98,79	98,78	63
eqmPzjjzrHk	Todos	0	0	0	100	100	100	100	73
enJGMKaiFiI	XGB	11,07	0	5,53	100	90,03	94,75	94,46	80
fIewI06H8u8	XGB	18,38	27,43	22,9	72,56	79,78	76	77,09	90
eerZKZFi2kY	RF XGB	33,44	0	16,72	100	74,93	85,67	83,27	120
dlj7Igoq3HQ	NBG	0	3,94	1,97	96,05	100	97,98	98,02	161
fDAVsmA3HUY	NBG	0	0,13	,06	99,86	100	99,93	99,93	210
f0ttzpoZyeA	NBG	0	4	2	95,99	100	97,95	97,99	225
eoWIhgawcZ0	NBG	0	4,4	2,2	95,59	100	97,74	97,69	253
fhw9jzhvkGs	XGB	2,43	0	1,21	100	97,61	98,79	98,78	270

Para a verificação dinâmica em EC foi possível encontrar um algoritmo com F1 a partir de 90%, para 47,82% dos usuários. Indicando que assim como aconteceu para a verificação estática a características relacionadas ao tamanho do dedo são importantes para criação de um modelo com boa performance. Os resultados dos melhores algoritmos por usuário para verificação dinâmica em EC são detalhados nas Tabelas 4.32, 4.33 e 4.34. .

Tabela 4.32: Detalhes dos melhores algoritmos VD Escopo C C1.

Usuário	Melhores Algoritmos VD EC C1								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	SVM	60,09	32,39	46,24	67,60	52,94	59,38	53,75	30
dUbEbDq40fM	NBB	39,77	2,52	21,14	97,47	71,02	82,17	78,85	36
fUB30EtiU0Q	XGB	0,99	0	0,49	100	99,01	99,5	99,5	36
cWN_XjnNRDw	NBB	27,19	2,44	14,81	97,55	78,2	86,81	85,18	42
ffdzWINClJ4	NBB	25	0	12,5	100	80	88,88	87,5	46
dFOtPe4f8Xc	XGB	8,3	16,02	12,16	83,97	90,99	87,34	87,83	65
cBc3b9Cv4X0	XGB	59,24	0	29,62	100	62,79	77,14	70,37	51
ev9fChXnR3I	SVM	24,1	49,43	36,77	50,56	67,71	57,89	63,22	48
fRA_pBm0ks4	NBB	35,58	0	17,79	100	73,75	84,89	82,2	54
dyRTk2BUAeo	NBB	3,54	47,76	25,65	52,23	93,64	67,05	74,34	53
cFxPtyX-07w	NBG	0	5,5	2,75	94,49	100	97,16	97,24	57
dVGmimRO7bE	NBG	57,14	0	28,57	100	63,63	77,77	71,42	57
fGTU-LDm8uM	NBG	59,78	0	29,89	100	62,58	76,98	70,1	60
fmVXDwdw20Q	SVM	3,19	18,65	10,92	81,34	96,22	88,16	89,07	63

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
eqmPzjjzrHk	RF	0,8	5,28	3,04	94,71	99,15	96,88	96,95	73
enJGMKaiFiI	GB	0,12	26,51	13,32	73,48	99,83	84,65	86,67	80
fIewI06H8u8	NBG	66,44	0	33,22	100	60,08	75,06	66,77	90
eerZKZFi2kY	NBB	13,12	0	6,56	100	88,39	93,84	93,43	120
dlj7Igoq3HQ	NBG	45,43	0,97	23,2	99,02	68,54	81,01	76,79	161
fDAVsmA3HUY	NBG	43,06	46,53	44,79	53,46	55,39	54,41	55,2	210
f0ttzpoZyeA	NBG	18,53	0	9,26	100	84,36	91,51	90,73	225
eoWIhgawcZ0	NBG	0	0,02	0,01	99,97	100	99,98	99,98	253
fhw9jzhvkGs	GB	61,68	0	30,84	100	61,85	76,42	69,15	270

Tabela 4.33: Detalhes dos melhores algoritmos VD Escopo C C2.

Usuário	Melhores Algoritmos VD EC C2								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dFOtPe4f8Xc	GB	16,18	20,23	18,2	79,76	83,13	81,41	81,79	65
cBc3b9Cv4X0	XGB	48,75	0	24,37	100	67,22	80,39	75,62	51
ev9fChXnR3I	NBB	99,3	0	49,65	100	50,17	66,82	50,34	48
fRA_pBm0ks4	SVM	30,24	0,71	15,48	99,28	76,64	86,51	84,51	54
dyRTk2BUAeo	NBG	0,28	0	0,14	100	99,71	99,85	99,85	53
cFxPtyX-07w	NBG	0	13,78	6,89	86,21	100	92,59	93,1	57
dVGmimRO7bE	XGB	78,36	0	39,18	100	56,06	71,84	60,81	57
fGTU-LDm8uM	RF GB XGB	51,04	0	25,52	100	66,2	79,66	74,47	60
fmVXDwdw20Q	SVM	4,15	0	2,07	100	96	97,96	97,92	63
eqmPzjjzrHk	NBB	20,19	0,32	10,25	99,67	83,15	90,66	89,74	73
enJGMKaiFiI	GB	6,95	0	3,47	100	93,49	96,63	96,52	80
fIewI06H8u8	NBG	72	0	36	100	58,13	73,52	63,99	90
eerZKZFi2kY	SVM	61,43	0	30,71	100	61,94	76,5	69,28	120
dlj7Igoq3HQ	GB	10,38	0,16	5,27	99,83	90,57	94,97	94,72	161
fDAVsmA3HUY	NBG	48,54	8,54	28,54	91,45	65,32	76,21	71,45	210
f0ttzpoZyeA	NBG	21,64	0	10,82	100	82,2	90,23	89,17	225
eoWIhgawcZ0	NBB	3,15	7,98	5,56	92,01	97,22	94,54	94,21	253
fhw9jzhvkGs	RF	27,96	0,41	14,19	99,58	78,07	87,52	85,8	270

Tabela 4.34: Detalhes dos melhores algoritmos VD Escopo C C3.

Usuário	Melhores Algoritmos VD EC C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG	95	0	47,5	100	51,28	67,79	52,5	60
fmVXDwdw20Q	RF	3,83	0	1,91	100	96,3	98,11	98,08	63
eqmPzjjzrHk	NBB	33,62	1,76	17,69	98,23	74,49	84,73	82,3	73
enJGMKaiFiI	XGB	0,71	0	0,35	100	99,29	99,64	99,64	80
fIewI06H8u8	SVM	36,35	0	18,17	100	73,34	84,61	81,82	90
eerZKZFi2kY	GB	62,82	0	31,41	100	61,41	76,09	68,58	120
dlj7Igoq3HQ	RF	72,04	0	36,02	100	58,12	73,51	63,97	161
fDAVsmA3HUY	NBG	65,87	0	32,93	100	60,28	75,22	67,06	210
f0ttzpoZyeA	NBG RF	24,8	0	12,4	100	80,12	88,96	87,59	225
eoWIhgawcZ0	NBB	14,29	3,53	8,91	96,46	88,11	92,10	91,34	253
fhw9jzhvkGs	GB	8,43	0,43	4,43	99,56	92,19	95,73	95,56	270

Para a verificação dinâmica em ED foi possível encontrar um algoritmo com F1 a partir de 90%, para 56,52% dos usuários. E assim como ocorreu para os modelos de verificação estática em ED, neste escopo foi possível encontrar modelos com F1 satisfatória para usuários para os quais não foi possível encontrar F1 satisfatório em EA. Indicando que estes podem ser escopos complementares também na verificação dinâmica. Os resultados dos melhores algoritmos por usuário para verificação dinâmica em ED são detalhados nas Tabelas 4.35, 4.36 e 4.37.

Tabela 4.35: Detalhes dos melhores algoritmos VD Escopo D C1.

Usuário	Melhores Algoritmos VD ED C1								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	RF	2,26	49,83	26,04	50,16	18,53	27,12	97,25	30
dUbEbDq40fM	RF	0,28	0	0,14	100	89,13	94,25	99,71	36
fUB30EtiU0Q	RF	0,98	8,10	4,54	91,89	51,68	66,16	98,93	36
cWN_XjnNRDw	RF	0,99	34,82	17,91	65,17	69,33	67,19	97,86	42
ffdzWINClJ4	RF	1,61	51,61	26,61	48,38	36,64	41,70	97,43	46
dFOtPe4f8Xc	RF	1,54	52,98	27,26	47,01	27,94	35,05	97,81	65
cBc3b9Cv4X0	RF	0,61	69,01	34,81	30,98	54,96	39,63	97,75	51
ev9fChXnR3I	RF	1,55	41,84	21,69	58,15	58,21	58,18	96,99	48
fRA_pBm0ks4	RF	0,06	0	0,03	100	97,57	98,77	99,93	54
dyRTk2BUAeo	RF	0,58	73,10	36,84	26,89	62,06	37,52	96,93	53
cFxPtyX-07w	RF	0,87	0	0,43	100	93,31	96,54	99,21	57
dVGmimRO7bE	RF	1,81	26,67	14,24	73,32	49,75	59,28	97,59	57
fGTU-LDm8uM	RF	2,35	79,04	40,19	21,95	22,36	22,15	95,37	60

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fmVXDwdw20Q	RF	7,02	17,74	12,38	82,25	28,70	45,55	92,61	63
eqmPzjjzrHk	RF	0	22,15	11,07	77,84	100	87,54	98,61	73
enJGMKaiFiI	RF	0,90	80	40,45	20	39	26,44	96,86	80
fIewI06H8u8	RF	0,44	25,17	12,80	74,82	88,88	81,24	98,44	90
eerZKZFi2kY	RF	3,20	72,99	38,09	27	28,29	27,63	93,67	120
dlj7Igoq3HQ	RF	1,81	60,47	31,14	39,52	69,27	50,33	92,68	161
fDAVsmA3HUY	RF	5,18	48,28	26,73	51,71	39,95	45,08	92,11	210
f0ttzpoZyeA	RF	1,90	58,01	29,96	41,98	44,89	43,39	96,08	225
eoWIhgawcZ0	RF	0,47	0	0,23	100	97,48	98,72	99,60	253
fhw9jzhvkGs	RF	0,64	87,62	44,13	12,37	49,43	19,79	95,16	270

Tabela 4.36: Detalhes dos melhores algoritmos VD Escopo D C2.

Usuário	Melhores Algoritmos VD ED C2								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dFOtPe4f8Xc	RF	1,07	1,47	1,27	98,52	47,07	63,70	98,91	65
cBc3b9Cv4X0	RF	0,09	99,53	49,81	0,46	9,52	0,88	97,88	51
ev9fChXnR3I	RF	0	67,36	33,68	32,63	99,46	49,14	98,18	48
fRA_pBm0ks4	RF	0	0	0	100	100	100	100	54
dyRTk2BUAeo	RF	0	59,34	29,67	40,65	100	57,80	98,29	53
cFxPtyX-07w	RF	0	0	0	100	100	100	100	57
dVGmimRO7bE	RF	0	0	0	100	100	100	100	57
fGTU-LDm8uM	RF	0,78	0	0,39	100	77,10	87,07	99,23	60
fmVXDwdw20Q	RF	3,24	1,82	2,53	98,17	66,63	79,38	96,84	63
eqmPzjjzrHk	RF	0	0	0	100	100	100	100	73
enJGMKaiFiI	RF	1,69	3,57	2,63	96,42	65,02	77,67	98,24	80
fIewI06H8u8	RF	2,04	3,49	2,77	96,5	71,83	82,35	97,88	90
eerZKZFi2kY	RF	0,23	31,95	16,09	68,04	94,98	79,28	97,85	120
dlj7Igoq3HQ	RF	5,01	1,79	3,40	98,20	72,94	83,70	95,37	161
fDAVsmA3HUY	RF	0,97	19,95	10,46	80,04	87,48	83,6	97,54	210
f0ttzpoZyeA	RF	0,11	37,10	18,61	62,89	95,97	75,99	98,29	225
eoWIhgawcZ0	RF	0,9	0	0,45	100	96,9	98,42	99,29	253
fhw9jzhvkGs	RF	0	70,85	35,42	29,14	100	45,13	99,33	270

Tabela 4.37: Detalhes dos melhores algoritmos VD Escopo D C3.

Usuário	Melhores Algoritmos VD ED C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG	0,02	0	0,01	100	98,75	99,37	99,97	60
fmVXDwdw20Q	NBG	1	0	0,5	100	67,37	80,50	99,01	63
eqmPzjjzrHk	NBG	0,06	0	0,03	100	98,93	99,46	99,93	73
enJGMKaiFiI	NBG	0,64	0,71	0,67	99,28	82,64	90,20	99,35	80
fIewI06H8u8	NBG	0	14,67	7,33	85,32	100	92,07	99,24	90
eerZKZFi2kY	RF	0,99	24,07	12,53	75,92	87,83	81,44	97,01	120
dlj7Igoq3HQ	NBG	0	6,34	3,17	93,65	100	96,72	98,98	161
fDAVsmA3HUY	NBG	0,84	9,40	5,12	90,59	92,79	91,68	98,23	210
f0ttzpoZyeA	NBG	0	0	0	100	100	100	100	225
eoWIhgawcZ0	NBG	0	0,04	0,022	99,95	100	99,97	99,98	253
fhw9jzhvkGs	NBG	0,32	0,22	0,27	99,77	96,98	98,36	99,68	270

Para a verificação dinâmica em EE foi possível encontrar um algoritmo com F1 a partir de 90%, para 34,79% dos usuários, indicando a importância dos dados de sensores para a definição de modelo de verificação dinâmica também. Os resultados dos melhores algoritmos por usuário para verificação dinâmica em EE são detalhados nas Tabelas 4.38, 4.39 e 4.40.

Tabela 4.38: Detalhes dos melhores algoritmos VD Escopo E C1.

Usuário	Melhores Algoritmos VD EE C1								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	NBG	0,30	58,52	29,41	41,47	57,67	48,24	99,11	30
dUbEbDq40fM	NBG	0,01	5,72	2,87	94,27	99,22	96,68	99,85	36
fUB30EtiU0Q	NBG	0,615	8,46	4,54	91,53	87,31	89,37	99,03	36
cWN_XjnNRDw	NBG	0	7,12	3,56	92,87	100	96,30	99,76	42
ffdzWINClJ4	NBG	0,34	15,58	7,96	84,41	85,83	85,11	99,28	46
dFOtPe4f8Xc	NBG	0,12	85,59	42,86	14,40	58,24	23,09	98,81	65
cBc3b9Cv4X0	NBG	1,21	98,56	49,84	1,43	2,73	1,88	96,51	51
ev9fChXnR3I	NBG	0,04	74,01	37,03	25,98	98,99	41,65	88,72	48
fRA_pBm0ks4	NBG	1,44	68,60	35,02	31,39	39,76	35,09	96,58	54
dyRTk2BUAeo	NBG	0,54	54,47	27,50	45,52	25,80	35,94	99,23	53
cFxPtyX-07w	NBG	1,04	38,71	19,87	61,28	87,53	72,09	94,92	57
dVGmimRO7bE	NBG	0,58	14,12	7,35	85,87	77,77	81,62	99,09	57
fGTU-LDm8uM	NBG	5,14	72,13	38,64	27,86	16,43	20,67	92,50	60
fmVXDwdw20Q	NBG	1,57	97,12	49,34	2,87	3,34	3,09	96,64	63
eqmPzjjzrHk	NBG	3,75	48,12	25,93	51,87	28,15	36,50	95,02	73
enJGMKaiFiI	NBG	3,29	46,94	25,12	53,05	18,88	27,84	96,08	80

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fIewI06H8u8	NBG	8,41	30,95	19,68	69,04	34,90	46,36	90,19	90
eerZKZFi2kY	NBG	4,22	86,05	45,14	13,94	10,28	11,83	93,02	120
dlj7Igoq3HQ	NBG	2,80	30,94	16,87	69,05	71,33	70,17	94,60	161
fDAVsmA3HUY	NBG	0,02	83,21	41,61	16,78	96,80	28,61	96,96	210
f0ttzpoZyeA	NBG	0,06	0	0,03	100	99,07	99,53	99,94	225
eoWIhgawcZ0	NBG	0	94,35	47,17	5,64	100	10,68	95,86	253
fhw9jzhvkGs	NBG	1,80	35,13	18,47	64,86	28,79	39,88	97,82	270

Tabela 4.39: Detalhes dos melhores algoritmos VD Escopo E C2.

Usuário	Melhores Algoritmos VD EE C2								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dFOtPe4f8Xc	NBG	0,29	75	37,64	25	45	32,27	98,98	65
cBc3b9Cv4X0	NBG	3,08	94,66	48,87	5,33	3,46	4,20	95,05	51
ev9fChXnR3I	NBG	0,91	97,71	49,31	2,28	6,46	3,37	96,48	48
fRA_pBm0ks4	NBG	0,5	2,66	1,58	97,33	79,28	87,39	99,45	54
dyRTk2BUAeo	NBG	2,37	16,93	9,38	83,60	51,10	63,43	97,22	53
cFxPtyX-07w	NBG	1,31	18,24	9,69	81,75	87,08	84,33	97,39	57
dVGmimRO7bE	NBG	0	0	0	100	100	100	100	57
fGTU-LDm8uM	NBG	2,85	71,40	37,13	28,59	21,01	24,22	95,36	60
fmVXDwdw20Q	NBG	10,53	29,79	20,16	70,20	17,20	27,64	88,87	63
eqmPzjjzrHk	NBG	0,24	0	0,12	100	96,15	98,04	99,76	73
enJGMKaiFiI	NBG	0,03	89,41	44,72	10,58	89,87	18,93	97,12	80
fIewI06H8u8	NBG	1,52	11,14	6,33	88,85	75,92	81,88	97,98	90
eerZKZFi2kY	NBG	0	94,21	47,10	5,78	100	10,93	94,30	120
dlj7Igoq3HQ	NBG	3,24	30,84	17,04	69,15	74,53	71,74	93,41	161
fDAVsmA3HUY	NBG	10,24	28,63	19,43	71,36	37,18	48,89	88,31	210
f0ttzpoZyeA	NBG	6,85	70,03	38,44	29,96	16,42	21,22	90,43	225
eoWIhgawcZ0	NBG	0,41	45,61	23,01	54,38	97,36	69,78	89,57	253
fhw9jzhvkGs	NBG	0,04	89,02	44,53	10,97	94,11	19,65	94,44	270

Tabela 4.40: Detalhes dos melhores algoritmos VD Escopo E C3.

Usuário	Melhores Algoritmos VD EE C3								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fGTU-LDm8uM	GB	0	32,35	16,17	67,64	100	80,70	99,45	60
fmVXDwdw20Q	GB	0,04	97,90	48,97	2,09	50	4,01	97,96	63
eqmPzjjzrHk	GB	0	0	0	100	100	100	100	73
enJGMKaiFiI	GB	0	0	0	100	100	100	100	80

Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fIewI06H8u8	GB	3,03	4,93	3,98	95,06	63,05	75,82	96,87	90
eerZKZFi2kY	GB	0,27	70,58	35,43	29,41	90,86	44,44	93,66	120
dlj7Igoq3HQ	GB	0,21	14,12	7,16	85,87	98,67	91,83	97,56	161
fDAVsmA3HUY	GB	1,76	92,57	47,16	7,42	33,73	12,17	88,46	210
f0ttzpoZyeA	GB	2,05	99,00	50,53	0,99	2,83	1,47	92,42	225
eoWIhgawcZ0	GB	3,60	2,20	2,90	97,79	92,45	95,04	96,82	253
fhw9jzhvkGs	GB	7,35	32,32	19,83	67,67	48,77	56,69	90,30	270

Para a verificação dinâmica em EF foi possível encontrar um algoritmo com F1 a partir de 90%, para 34,78% dos usuários, indicando a importância dos dados de tamanho do dedo para a definição de modelo de verificação dinâmica também. Os resultados dos melhores algoritmos por usuário para verificação dinâmica em EF são detalhados nas Tabelas 4.41, 4.42, 4.43.

Tabela 4.41: Detalhes dos melhores algoritmos VD Escopo F C1.

Usuário	Melhores Algoritmos VD EF C1								
Identificação	ALG	FAR	FRR	EER	REC	PRC	F1	AC	QTD
cwdNwCqtFAs	NBG	0	49,82	24,91	50,16	100	66,81	99,49	30
dUbEbDq40fM	NBG	0	57,85	28,92	42,14	100	59,29	98,65	36
fUB30EtiU0Q	NBG	0	76,57	38,28	23,42	100	37,95	99,13	36
cWN_XjnNRDw	NBG	0	69,75	34,87	30,24	100	46,44	97,66	42
ffdzWINClJ4	NBG	0,84	0	0,42	100	69,5	82	99,16	46
dFOtPe4f8Xc	NBG	0	69,56	34,78	30,43	100	46,66	99,12	65
cBc3b9Cv4X0	NBG	10,42	43,18	26,8	56,81	11,7	19,41	88,79	51
ev9fChXnR3I	NBG	0,77	80,17	40,47	19,82	48,17	28,18	96,36	48
fRA_pBm0ks4	NBG	0	0,27	0,13	99,72	100	99,86	99,99	54
dyRTk2BUAeo	NBG	0	80,37	40,18	19,62	100	32,8	97,24	53
cFxPtyX-07w	NBG	0	3,75	1,87	96,24	100	98,08	99,59	57
dVGmimRO7bE	NBG	3,43	67,04	35,23	32,95	19,04	24,13	95,04	57
fGTU-LDm8uM	NBG	0	79,63	39,81	20,36	100	33,83	97,61	60
fmVXDwdw20Q	NBG	2,42	34,25	18,33	65,74	48,26	55,66	96,51	63
eqmPzjjzrHk	NBG	3,89	0	1,94	100	63,03	77,32	96,34	73
enJGMKaiFiI	NBG	0	98,9	49,45	1,09	100	2,15	97,21	80
fIewI06H8u8	NBG	8,6	49,96	29,28	50,03	21,54	30,11	89,52	90
eerZKZFi2kY	NBG	12,26	87,64	49,95	12,35	4,5	6,59	84,36	120
dlj7Igoq3HQ	NBG	0,85	76,5	38,68	23,49	73,91	35,65	92,04	161
fDAVsmA3HUY	NBG	0	90,19	45,09	9,8	100	17,86	94,35	210
f0ttzpoZyeA	NBG	0	58,01	29	41,98	100	59,13	97,92	225
eoWIhgawcZ0	NBG	0	0	0	100	100	100	100	253
fhw9jzhvkGs	NBG	0,38	4,52	2,45	95,47	92,59	94,01	99,41	270

Tabela 4.42: Detalhes dos melhores algoritmos VD Escopo F C2.

Usuário	Melhores Algoritmos VD EF C2								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	SVM	0,1	61,76	30,93	38,23	77,22	51,14	99,29	65
cBc3b9Cv4X0	SVM	1,32	95,35	48,34	4,64	6,75	5,5	96,75	51
ev9fChXnR3I	SVM	4,66	21,75	13,21	78,24	31,67	45,09	94,87	48
fRA_pBm0ks4	SVM	0,07	0	0,03	100	96,27	98,09	99,92	54
dyRTk2BUAeo	SVM	1,59	57,37	29,48	42,62	44,14	43,36	96,79	53
cFxPtyX-07w	SVM	0	0	0	100	100	100	100	57
dVGmimRO7bE	SVM	5,38	31,09	18,23	68,9	17,99	28,53	94,18	57
fGTU-LDm8uM	SVM	1,16	61,74	31,45	38,25	46,66	42,04	97,26	60
fmVXDwdw20Q	SVM	0,29	71,49	35,89	28,5	86,37	42,86	95,29	63
eqmPzjjzrHk	SVM	0,4	4,24	2,32	95,75	93,54	94,63	99,37	73
enJGMKaiFiI	SVM	3,72	27,57	15,64	72,42	38,87	50,59	95,51	80
fIewI06H8u8	SVM	1,33	95,76	48,54	4,23	14,64	6,57	93,82	90
eerZKZFi2kY	SVM	1,32	11,4	6,36	88,59	81,17	84,72	98,06	120
dlj7Igoq3HQ	SVM	2,4	17,96	10,18	82,03	82,42	82,23	95,71	161
fDAVsmA3HUY	SVM	2,84	65,28	34,06	34,71	50,92	41,29	92,26	210
f0ttzpoZyeA	SVM	1,04	38,74	19,89	61,25	72,56	66,42	97,33	225
eoWIhgawcZ0	SVM	0,2	0,04	0,12	99,95	99,27	99,61	99,83	253
fhw9jzhvkGs	SVM	0	0	0	100	100	100	100	270

Tabela 4.43: Detalhes dos melhores algoritmos VD Escopo F C3.

Usuário	Melhores Algoritmos VD EF C3								
Identificação	**ALG**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG	2,15	65,96	34,05	34,03	21,31	26,21	96,77	60
fmVXDwdw20Q	NBG	1,01	0	0,5	100	67,05	80,27	99	63
eqmPzjjzrHk	NBG	0,22	0	0,11	100	96,54	98,24	99,78	73
enJGMKaiFiI	NBG	2,82	10,18	6,5	89,81	49,47	63,8	96,55	80
fIewI06H8u8	NBG	1,41	77,36	39,38	22,63	46,61	30,47	94,66	90
eerZKZFi2kY	NBG	3,69	22,67	13,18	77,32	66,36	71,42	94,66	120
dlj7Igoq3HQ	NBG	0	7,37	3,68	92,62	100	96,17	98,82	161
fDAVsmA3HUY	NBG	1,77	54,43	28,1	45,56	75,57	56,84	92,55	210
f0ttzpoZyeA	NBG	0	7,96	3,98	92,03	100	95,85	99,54	225
eoWIhgawcZ0	NBG	0	0,04	0,02	99,95	100	99,97	99,98	253
fhw9jzhvkGs	NBG	1,38	0,22	0,8	99,77	88,18	93,62	98,72	270

Diante dos resultados capturados nos escopos de verificação dinâmica de EA a EF, foi definido

que o *framework*, assim como para a verificação estática incorporará os escopos ED, EA e EB, nesta ordem de precedência, pois foi observado que apesar do escopo B e D terem sido capazes de encontrar o F1 desejado para a mesma porcentagem de usuários o ED é complementar ao EA, assim como aconteceu para a verificação estática. Os escopos EA e EB também só serão utilizados caso não seja possível encontrar o F1 de pelo menos 90% para o usuário em ED, assim como para VE. O resultado para o passo 4 do *framework* proposto, encontrar um modelo com pelo menos 90% para F1, com a união dos escopos EA, EB e ED é demonstrada nas Tabelas 4.44, 4.45 e 4.46.

Tabela 4.44: Detalhes dos melhores algoritmos candidatos VD *Framework* C1.

Usuário	**Melhores Algoritmos candidatos VD *Framework* C1**								
Identificação	**ALG (E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
cwdNwCqtFAs	NBG (EB)	0	6,57	3,28	93,42	100	96,6	96,71	30
dUbEbDq40fM	RF (ED)	0,28	0	0,14	100	89,13	94,25	99,71	36
fUB30EtiU0Q	XGB (EA)	0,99	0	0,49	100	99,01	99,5	99,5	36
cWN_XjnNRDw	NBG (EB)	23,32	0	11,66	100	81,08	89,55	88,33	42
ffdzWINClJ4	NBG (EB)	32,4	0	16,2	100	75,52	86,05	83,8	46
dFOtPe4f8Xc	NBG (EB)	0	1,78	0,89	98,21	100	99,1	99,1	65
cBc3b9Cv4X0	XGB (EB)	0	0,15	0,07	99,84	100	99,92	99,92	51
ev9fChXnR3I	NBG (EB)	93,6	0,11	46,86	99,88	51,62	68,06	53,13	48
fRA_pBm0ks4	RF (ED)	0,06	0	0,03	100	97,57	98,77	99,93	54
dyRTk2BUAeo	GB (EA)	32,34	5,38	18,96	94,61	74,4	83,3	81,03	53
cFxPtyX-07w	RF (ED)	0,87	0	0,43	100	93,31	96,54	99,21	57
dVGmimRO7bE	NBG (EB)	17,69	0	8,84	100	84,96	91,87	91,15	57
fGTU-LDm8uM	NBG (EB)	55,97	0,39	28,18	99,6	64,02	77,94	71,81	60

Identificação	ALG (E)	FAR	FRR	EER	REC	PRC	F1	AC	QTD
fmVXDwdw20Q	NBG (EB)	0	8,83	4,41	91,16	100	95,37	95,58	63
eqmPzjjzrHk	NBB (EA)	0	0,14	0,07	99,85	100	99,92	99,92	73
enJGMKaiFiI	XGB (EA)	27,01	0,86	13,94	99,13	78,58	87,67	86,05	80
fIewI06H8u8	NBG (EB)	37,37	1,61	19,49	98,38	72,47	83,46	80,5	90
eerZKZFi2kY	XGB (EA)	0,77	49,8	25,28	50,19	98,48	66,49	74,71	120
dlj7Igoq3HQ	NBG (EB)	0	3,34	1,67	96,65	100	98,29	98,32	161
fDAVsmA3HUY	NBG (EB)	0	0,11	0,05	99,88	100	99,94	99,94	210
f0ttzpoZyeA	NBG (EA)	18,53	0	9,26	100	84,36	91,51	90,73	225
eoWIhgawcZ0	RF (ED)	0,47	0	0,23	100	97,48	98,72	99,60	253
fhw9jzhvkGs	SVM (EB)	11,27	21,07	16,17	78,92	87,5	82,99	83,82	270

Tabela 4.45: Detalhes dos melhores algoritmos VD *Framework* C2.

Usuário	Melhores Algoritmos VD *Framework* C2								
Identificação	**ALG (E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
dFOtPe4f8Xc	RF (EB)	0	3,46	1,73	96,53	100	98,23	98,26	65
cBc3b9Cv4X0	NBG (EB)	0	0,31	0,15	99,68	100	99,84	99,84	51
ev9fChXnR3I	NBG (EB)	92,36	0	46,18	100	51,98	68,4	53,81	48
fRA_pBm0ks4	RF (ED)	0	0	0	100	100	100	100	54
dyRTk2BUAeo	SVM (EA)	0,28	0	0,14	100	99,71	99,85	99,85	53
cFxPtyX-07w	RF (ED)	0	0	0	100	100	100	100	57

Identificação	ALG(E)	FAR	FRR	EER	REC	PRC	F1	AC	QTD
dVGmimRO7bE	RF (ED)	0	0	0	100	100	100	100	57
fGTU-LDm8uM	RF (ED)	0,78	0	0,39	100	77,10	87,07	99,23	60
fmVXDwdw20Q	SVM (EA)	2,07	0	1,03	100	97,96	98,97	98,96	63
eqmPzjjzrHk	RF (ED)	0	0	0	100	100	100	100	73
enJGMKaiFiI	XGB (EA)	21,21	0	10,6	100	82,49	90,4	89,39	80
fIewI06H8u8	GB (EB)	21,95	0,8	11,87	98,19	81,72	89,21	88,12	90
eerZKZFi2kY	GB XGB (EA)	31,87	0	15,89	100	75,88	86,28	84,1	120
dlj7Igoq3HQ	GB (EA)	16,71	0	8,35	100	85,67	92,28	91,64	161
fDAVsmA3HUY	NBG (EA)	0	5,44	2,72	94,55	100	97,19	97,27	210
f0ttzpoZyeA	NBG RF GB (EA)	21,64	0	10,82	100	82,2	90,23	89,17	225
eoWIhgawcZ0	RF (ED)	0,9	0	0,45	100	96,9	98,42	99,29	253
fhw9jzhvkGs	RF (EA)	3,5	13,93	8,72	86,06	96,08	90,79	91,27	270

Tabela 4.46: Detalhes dos melhores algoritmos VD *Framework* C3.

Usuário	Melhores Algoritmos VD *Framework* C3								
Identificação	**ALG(E)**	**FAR**	**FRR**	**EER**	**REC**	**PRC**	**F1**	**AC**	**QTD**
fGTU-LDm8uM	NBG (ED)	0,02	0	0,01	100	98,75	99,37	99,97	60
fmVXDwdw20Q	SVM (EA)	1,04	0	0,52	100	98,96	99,48	99,47	63
eqmPzjjzrHk	NBG (ED)	0,06	0	0,03	100	98,93	99,46	99,93	73

Identificação	ALG(E)	FAR	FRR	EER	REC	PRC	F1	AC	QTD
enJGMKaiFiI	NBG (ED)	0,64	0,71	0,67	99,28	82,64	90,20	99,35	80
fIewI06H8u8	NBG (ED)	0	14,67	7,33	85,32	100	92,07	99,24	90
eerZKZFi2kY	XGB (EA)	33,44	0	16,72	100	74,93	85,67	83,27	120
dlj7Igoq3HQ	NBG (ED)	0	6,34	3,17	93,65	100	96,72	98,98	161
fDAVsmA3HUY	NBG (ED)	0,84	9,40	5,12	90,59	92,79	91,68	98,23	210
f0ttzpoZyeA	NBG (ED)	0	0	0	100	100	100	100	225
eoWIhgawcZ0	NBG (ED)	0	0,04	0,022	99,95	100	99,97	99,98	253
fhw9jzhvkGs	NBG (ED)	0,32	0,22	0,27	99,77	96,98	98,36	99,68	270

Conforme o resultado detalhado na Tabela 4.44, foi possível encontrar um algoritmo com F1 a partir de 90% para 60,86% dos usuários no cenário 1, para 77,77% dos usuários no cenário 2, conforme Tabela 4.45, e para 90,9% dos usuários no cenário 3, conforme detalhado na Tabela 4.46.

Com a abordagem proposta, unindo os escopos EA, EB e ED, no *framework* foi possível encontrar em um ou mais cenários, um algoritmo com F1 pretendido para 86,95% os usuários, ficando apenas 4 usuários, dos quais 3 não ofereceram amostras suficientes para participar do cenário 3, sem um modelo com a F1 pretendido para a verificação dinâmica. Sendo o escopo D responsável por solucionar para 17% dos usuários em C1, 27% em C2 e 81,81% em C3, o Escopo A para 13,04% dos usuários em C1, 38,88% em C2 e 9,09% em C3 e o Escopo B para 30,43% dos usuários em C1, 16,66% em C2 e 0% em C3. Diferentemente da verificação estática onde a resolução da busca da F1 pretendido ficou muito concentrada em ED, no caso da verificação dinâmica houve uma maior distribuição entre os escopos, indicando que é mais complexa a tarefa de encontrar um modelo com a F1 pretendido para VD, que envolve um conjunto de telas, algumas que contam com apenas um toque na tela para a interação, e não apenas uma como em VE.

O resultado para o passo 5 do *framework* proposto, modelos com FAR de até 10% para os *templates* dos outros usuários, impostores, são detalhados na Tabela 4.47. Este é o último crivo que o modelo deve passar para ser considerado apto para a autenticação do usuário.

Tabela 4.47: Resultados finais algoritmos por cenário VD *Framework*.

Usuário	**VD *Framework* Resultado Final**								
Identificação	**ALG (E) C1**	**F1 C1**	**FAR_I C1**	**ALG (E) C2**	**F1 C2**	**FAR_I C2**	**ALG (E) C3**	**F1 C3**	**FAR_I C3**
cwdNwCqtFAs	NBG (EB)	96,6	0	—	—	—	—	—	—
dUbEbDq40fM	RF (ED)	94,25	0,01	—	—	—	—	—	—
fUB30EtiU0Q	XGB (EA)	99,5	53,64	—	—	—	—	—	—
cWN_XjnNRDw	—	—	—	—	—	—	—	—	—
ffdzWINClJ4	—	—	—	—	—	—	—	—	—
dFOtPe4f8Xc	NBG (EB)	99,1	0	RF (EB)	98,23	11,01	—	—	—
cBc3b9Cv4X0	XGB (EB)	99,92	11,16	NBG (EB)	99,84	15,16	—	—	—
ev9fChXnR3I	—	—	—	—	—	—	—	—	—
fRA_pBm0ks4	RF (ED)	98,77	10,13	RF (ED)	100	5,11	—	—	—
dyRTk2BUAeo	—	—	—	SVM (EA)	99,85	14,87	—	—	—
cFxPtyX-07w	RF (ED)	96,54	0,2	RF (ED)	100	0,14	—	—	—
dVGmimRO7bE	NBG (EB)	91,87	0	RF (ED)	100	0,5	—	—	—
fGTU-LDm8uM	—	—	—	—	—	—	NBG (ED)	99,37	0
fmVXDwdw20Q	NBG (EB)	95,37	8,71	SVM (EA)	98,97	25,38	SVM (EA)	99,48	23,41
eqmPzjjzrHk	NBB (EA)	99,92	32,13	RF (ED)	100	0,75	NBG (ED)	99,46	100
enJGMKaiFiI	—	—	—	XGB (EA)	90,4	52,6	NBG (ED)	90,2	0
fIewI06H8u8	—	—	—	—	—	—	NBG (ED)	92,07	0
eerZKZFi2kY	—	—	—	—	—	—	—	—	—
dlj7Igoq3HQ	NBG (EB)	98,29	8,27	GB (EA)	92,28	65,9	NBG (ED)	96,72	0

Identificação	ALG (E) C1	F1 C1	FAR_I C1	ALG (E) C2	F1 C2	FAR_I C2	ALG (E) C3	F1 C3	FAR_I C3
fDAVsmA3HUY	NBG (EB)	99,94	0	NBG (EA)	97,19	0	NBG (ED)	91,68	0
f0ttzpoZyeA	NBG (EA)	91,51	59,03	NBG (EA)	90,23	0	NBG (ED)	100	0
eoWIhgawcZ0	RF (ED)	98,72	0,17	RF (ED)	98,42	1,22	NBG (ED)	99,97	0
fhw9jzhvkGs	—	—	—	RF (EA)	90,79	72,04	NBG (ED)	98,36	0

Diante dos resultados demonstrados na Tabela 4.25, para o passo final do *framework* proposto, foi possível encontrar um modelo com F1 de pelo menos 90% e com FAR_I de até 10% para 69,56% dos usuários em VD, 16 usuários, sendo que 7 para os quais não foi possível encontrar um modelo que atendesse aos requisitos os usuários só ofereceram amostras suficientes para participar de até C2. Dos usuários que ofereçam amostras suficientes para participar de C3, apenas 1 não teve um bom modelo criado, indicando que para VD é necessário mais dados de treino para se construir um modelo com boa performance. Diante disso foi comprovado que é promissora a utilização de dados de interação pós login para autenticação de um usuário.

4.2.3 Média Resultados VE e VD

Nesta subseção serão feitas as análises dos resultados do *framework* proposto quantos aos resultados esperados, tendo com referência os resultados observados nos trabalhos listados na revisão da literatura. Todas as comparações são feitas para que o *framework* proposto possa ter um referencial mínimo de comparação de desempenho, mesmo que as características e métricas utilizadas em cada trabalho de revisão da literatura não sejam, exatamente, as mesmas utilizadas no *framework* proposto.

Conforme Tabelas 4.48, 4.49 e 4.50, onde são detalhados a acurácia, EER e F1 médios entre os cenários para VE, foi possível obter um resultado de até 98,25% de acurácia com o modelo proposto no *framework* para o cenário 3, mesmo que essa não seja a métrica considerada no *modelo proposto*, para efeito de comparação foi obtida uma acurácia maior que a descrita na revisão da literatura, onde o maior valor de acurácia relatado foi de 96%, para verificação estática em aplicação bancária *mobile* [36] e de 93,04% para verificação estática em [7]. Já o EER variou entre 4,57 e 1,88% e o F1 entre 95,32 e 97,05%, resultados médios superiores ao limiar definido em 90%, indicando que o modelo proposto conseguiu ter uma boa performance para a maioria dos usuários que participaram do experimento em VE.

Tabela 4.48: Acurácia média por cenário VE.

Acurácia média por cenário VE							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	89,44	87,12	86,64	98,85	96,82	97,61	95,81
C2	91,02	88,35	84,7	98,13	95,87	97,05	97,38
C3	90,77	87,29	87,55	98,33	96,06	97,96	98,25

Tabela 4.49: EER médio por cenário VE.

EER médio por cenário VE							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	10,14	11,45	13,51	7,3	19,18	11,95	4,57
C2	8,97	11,62	15,23	5,87	18,47	11,47	2,87
C3	9,21	12,63	12,49	4,59	13,23	6,46	1,88

Tabela 4.50: F1 médio por cenário VE.

F1 médio por cenário VE							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	90,4	88,24	87,91	83,79	59,59	70,58	95,32
C2	91,68	87,32	86,9	85,34	60,53	73,91	96,34
C3	91,44	83,7	87,55	87,9	71,62	84,58	97,05

Em relação a VD foi possível observar um EER de até 3,07% no cenário 3, inferior ao relatado na literatura para autenticação dinâmica em [8], com 4% entre semanas, conforme detalhado na Tabela 4.52. A acurácia variou entre 90,1 e 98%, conforme detalhado na Tabela 4.51. O F1 *score* variou entre 90,68 e 95,72, conforme detalhado na Tabela 4.53, valor médio superior ao limiar definido em 90% indicando que o modelo proposto apresentou boa performance para a maioria dos usuários.

Tabela 4.51: Acurácia média por cenário VD.

Acurácia média por cenário VD							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	80,29	85,72	80,27	96,89	96,5	96,32	90,1
C2	84,38	85,22	81,84	95,49	95,22	97,01	93,34
C3	84,86	90,88	80,76	99,21	95,77	97,22	98

Tabela 4.52: EER médio por cenário VD.

EER médio por cenário VD							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	21,15	14,31	19,71	21,63	25,9	26,55	9,85
C2	15,82	14,73	18,12	11,51	25,24	19,14	6,61
C3	15,15	9,09	19,24	2,69	21,1	11,84	3,07

Tabela 4.53: F1 médio por cenário VD.

F1 médio por cenário VD							
Cenário	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**Escopo proposto**
C1	81,71	87,09	81,95	55,18	47,92	50,17	90,68
C2	87,32	88,43	85,43	75,78	48,21	60,14	93,73
C3	88,07	90,35	85,13	93,61	60,19	73,89	95,72

A abordagem proposta do *framework*, apesar de agregar uma certa complexidade, pois é derivada da junção dos escopos EA, EB e ED, apresentou melhor EER médio entre VE e VD se comparado com o apresentado em [37], EER de 11,5% para verificação estática e dinâmica em aplicação bancária *mobile*. Sendo assim o modelo proposto atingiu o objetivo de ter melhor performance do que os trabalhos apresentados na revisão da literatura tanto para VE quanto para VD.

4.3 DETECÇÃO DE IMPOSTORES

Um modelo para ser utilizado em biometria comportamental tem que ser bom em classificar o usuário legítimo, mantendo um FAR e um FRR equilibrados e com uma taxa baixa, pois isso indica que o modelo é efetivo tanto na identificação do usuários legítimos quanto dos impostores. Conforme detalhado na seção anterior, para VE o EER variou entre 1,88 e 4,57%, e entre 3,07 e 9,85% para VD no *framework* proposto, se mantendo equilibrado e com valores baixos, principalmente se considerado apenas C3, 1,88% para VE e 3,07% para VD, entre semanas. E além deste equilíbrio e bons valores encontrados, tentando tornar os modelos encontrados, segundo a definição do *framework*, mais resilientes a impostores só foi considerado apto o modelo que obtivesse F1 a partir de 90%, e que também não obtivesse FAR maior que 10%, quando confrontado com os *templates* de todos os outros 50 usuários que participaram do experimento.

Para os usuários que conseguiram ter um modelo que passou por todos os crivos dos passos definidos do *framework*, e tiveram um FAR_I maior que 1%, em nenhum dos casos o modelo identificou em 100% um usuário impostor com um legítimo, essa valor de FAR_I maior que 1% foi um somatório de autenticações, dentro da margem do limiar aceito, de vetores de dados de

vários usuários impostores como um usuário legítimo, indicando que para os modelos criados, atendendo todos os requisitos estabelecidos, um usuário impostor só conseguirá se passar pelo legítimo no máximo em 10% das tentativas. Se levado em consideração os resultados da seção anterior, isso só aconteceria na minoria dos modelos e, se for julgado necessário, o *framework* pode ser ajustado para aceitar um FAR_I menor. Demonstrando que o *framework* proposto pode ser bem eficiente em identificar os impostores, com base nas características e escopos propostos, na utilização de uma aplicação bancária *mobile*, similar à desenvolvida neste experimento.

Mesmo com todas estas precauções, definidas nos requisitos de criação do modelo para o *framework*, ainda assim, um impostor pode ter acesso ao sistema sem que seja detectado, pois existem as margens de erro, e conforme demonstrado em [59], os dados de autenticação continua baseado em biometria comportamental podem sofrer ataques de imitação. Em [59], não foram levados em consideração os dados dos vários sensores disponíveis nos *smartphones*, diferentemente do experimento proposto neste trabalho. Quanto a ataques contra biometria comportamental *touch*, considerando dados de sensores, os autores em [60], sugerem que a consideração de dados de sensores pode ser um forte mecanismo de autenticação biométrica contra ataques práticos recentemente propostos.

4.3.1 Frequência dos algoritmos para VE e VD

Durante os experimentos a frequência dos algoritmos com F1 a partir de 90% variou bastante entre os escopos, devido às características utilizadas em cada escopo para criar os modelos. A Tabela 4.54 detalha a frequência para VE EA, onde o algoritmo NBB e NBG foram os que apresentaram maior frequência.

Tabela 4.54: Frequência algoritmos F1 a partir de 90% VE EA.

Frequência VE EA			
Usuário	**C1**	**C2**	**C3**
NBB	6	4	2
NBG	6	4	3
SVM	2	—	1
RF	2	—	—
GB	3	—	2
XGB	2	1	1

A Tabela 4.55 detalha a frequência para VE EB, onde foi mantida a maior frequência para os algoritmos NBB e NBG.

Tabela 4.55: Frequência algoritmos F1 a partir de 90% VE EB.

Frequência VE EB			
Usuário	**C1**	**C2**	**C3**
NBB	1	2	—
NBG	11	5	4
SVM	—	1	2
RF	—	2	—
GB	—	—	—
XGB	2	2	—

A Tabela 4.56 detalha a frequência para VE EC, onde foi mantida a maior frequência para os algoritmos NBB e NBG.

Tabela 4.56: Frequência algoritmos F1 a partir de 90% VE EC.

Frequência VE EC			
Usuário	**C1**	**C2**	**C3**
NBB	6	2	1
NBG	5	3	3
SVM	3	—	—
RF	1	—	1
GB	2	1	3
XGB	1	3	—

A Tabela 4.57 detalha a frequência para VE ED, onde o algoritmo que teve o melhor desempenho para todos os cenários foi o RF.

Tabela 4.57: Frequência algoritmos F1 a partir de 90% VE ED.

Frequência VE ED			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	—	—	—
SVM	—	—	—
RF	13	12	9
GB	—	—	—
XGB	—	—	—

A Tabela 4.58 detalha a frequência para VE EE, onde o algoritmo XGB apresentou maior frequência.

Tabela 4.58: Frequência algoritmos F1 apartir de 90% VE EE.

Frequência VE EE			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	—	—	—
SVM	—	—	—
RF	—	—	—
GB	—	—	—
XGB	6	4	5

A Tabela 4.59 detalha a frequência para VE EF, onde o algoritmo XGB apresentou maior frequência.

Tabela 4.59: Frequência algoritmos F1 a partir de 90% VE EF.

Frequência VE EF			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	—	—	—
SVM	—	—	—
RF	8	—	—
GB	—	—	—
XGB	—	6	7

A Tabela 4.60 detalha a frequência para VE para o *framework* proposto, onde o algoritmo RF apresentou maior frequência.

Tabela 4.60: Frequência algoritmos VE *Framework*.

Frequência VE *Framework*			
Usuário	**C1**	**C2**	**C3**
NBB	3	1	—
NBG	3	2	1
SVM	—	—	—
RF	12	12	10
GB	—	—	—
XGB	—	—	—

Conforme detalhado nas Tabelas 4.54, 4.55, 4.56, 4.57, 4.58 , 4.59 e 4.60 sobre as frequências dos melhores algoritmos para VE, na criação dos modelos por usuário de forma individual, EA, EB e EC, a maior frequência foi para os algoritmos NBB e NBG, já para os cenários com

um modelo compartilhado entre todos os usuários foi possível perceber a melhor performance para os algoritmos baseados em métodos *ensemble* e isso foi refletido também na frequência dos algoritmos para o *framework*, onde os resultados do escopo D são predominantes.

A Tabela 4.61 detalha a frequência dos algoritmos para VD EA, onde os algoritmos NBB e NBG apresentaram maior frequência.

Tabela 4.61: Frequência algoritmos F1 a partir de 90%VD EA.

Frequência VD EA			
Usuário	**C1**	**C2**	**C3**
NBB	2	1	1
NBG	2	3	—
SVM	—	3	1
RF	—	2	—
GB	—	2	1
XGB	1	1	1

A Tabela 4.62 detalha a frequência dos algoritmos para VD EB, onde os algoritmos NBB e NBG apresentaram maior frequência.

Tabela 4.62: Frequência algoritmos F1 a partir de 90%VD EB.

Frequência VD EB			
Usuário	**C1**	**C2**	**C3**
NBB	1	—	—
NBG	8	8	4
SVM	1	—	—
RF	—	—	—
GB	—	—	—
XGB	1	—	3

A Tabela 4.63 detalha a frequência dos algoritmos para VD EC, onde os algoritmos NBB e NBG apresentaram maior frequência.

Tabela 4.63: Frequência algoritmos F1 a partir de 90% VD EB.

Frequência VD EC			
Usuário	**C1**	**C2**	**C3**
NBB	1	2	1
NBG	4	3	—
SVM	—	1	—
RF	1	—	1

GB	—	2	1
XGB	1	—	1

A Tabela 4.64 detalha a frequência dos algoritmos para VD ED, onde o algoritmo RF apresentou maior frequência, mas diferente de VE ED no cenário 3 o algoritmo NBG foi o melhor, tendo assim nesse escopo uma distribuição das frequências entre os métodos ensemble e métodos baseados em probabilidades.

Tabela 4.64: Frequência algoritmos F1 a partir de 90% VD ED.

Frequência VD ED			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	—	—	9
SVM	—	—	—
RF	4	6	—
GB	—	—	—
XGB	—	—	—

A Tabela 4.65 detalha a frequência dos algoritmos para VD EE, onde o algoritmo NBG apresentou maior frequência, seguido do GB, novamente com uma distribuição de frequência entre os algoritmos *ensemble* e não *ensemble*.

Tabela 4.65: Frequência algoritmos F1 a partir de 90%VD EE.

Frequência VD EE			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	3	2	—
SVM	—	—	—
RF	—	—	—
GB	—	—	4
XGB	—	—	—

A Tabela 4.66 detalha a frequência dos algoritmos para VD EF, onde o algoritmo NBG apresentou maior frequência.

Tabela 4.66: Frequência algoritmos F1 a partir de 90% VD EF.

Frequência VD EF			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	5	—	5
SVM	—	5	—
RF	—	—	—
GB	—	—	—
XGB	—	—	—

A Tabela 4.67 detalha a frequência para VD para o *framework* proposto, onde o algoritmo NBG apresentou maior frequência.

Tabela 4.67: Frequência algoritmos VD *Framework*.

Frequência VD *Framework*			
Usuário	**C1**	**C2**	**C3**
NBB	—	—	—
NBG	6	2	8
SVM	—	—	—
RF	3	5	—
GB	—	—	—
XGB	—	—	—

Conforme detalhado nas Tabelas 4.61, 4.62, 4.63, 4.64, 4.65, 4.66 e 4.67 sobre as frequências dos algoritmos para VD, na criação dos modelos por usuário de forma individual, EA, EB e EC, a maior frequência foi encontrada para os algoritmos NBB e NBG, já para os cenários com um modelo compartilhado entre todos os usuários e para o *framework* foi possível perceber a melhor performance para os algoritmos baseados em métodos *ensemble*, apesar de haver uma distribuição mais equilibrada para as frequências em relação a VE, com destaque para o algoritmo NBG.

4.3.2 Hiperparâmetros Melhores Algoritmos

Os valores dos hiperparâmetros para os algoritmos baseados em *ensemble* e o SVM, variou entre os cenários em que estes foram encontrados como melhores algoritmos na busca pela F1 desejado, conforme será detalhado nesta seção.

Em VE EA, conforme detalhado na Tabela 4.68, para os algoritmos com hiperparâmetros, foram encontrados como os melhores 13 vezes entre todos os cenários, com frequência 3 para SVM, 2 para RF, 5 para GB e 3 para XGB. Para o SVM , quanto à análise dos hiperparâmetros, os *kernels* variaram entre linear e rbf, sempre com a classe 1 com peso maior. Para o RF o número

de estimadores, árvores de decisão que serão utilizadas, variou entre 20 e 30, a quantidade máxima de características por árvore de decisão ficou em 3. Para o GB o número de estimadores variou entre 10 e 100, a profundidade máxima das árvores ficou em 5. Para o XGB, a quantidade de estimadores ficou fixa em 20 para todos os usuários, com uma profundidade máxima para as árvores em 15.

Tabela 4.68: Hiperparâmetros algoritmos F1 a partir de 90% VE EA.

Hiperparâmetros algoritmos F1 a partir de 90% VE EA			
Usuário	**C1**	**C2**	**C3**
eHC7qNMAdCI	GB: {'learning_rate': 0.001, 'max_depth':5, 'max_features':7, 'min_samples_leaf': 0.2, 'min_samples_split': 0.3, 'n_estimators':30, 'random_state':0}	—	—
fUB30EtiU0Q	GB: {'learning_rate': 0.1, 'max_depth':5, 'max_features':3, 'min_samples_leaf': 0.2, 'min_samples_split': 0.45, 'n_estimators':75, 'random_state':0}	—	—

Usuário	C1	C2	C3
fRA_pBm0ks4	GB: {'learning_rate': 0.001, 'max_depth':5, 'max_features':3, 'min_samples_leaf': 0.2, 'min_samples_split': 0.4, 'n_estimators':10, 'random_state':0}	—	—
cFxPtyX-07w	SVM: {'C':0.001, 'class_weight': 0: 1, 1: 2, 'kernel':'linear'}	—	—
dyRTk2BUAeo	—	—	GB: {'learning_rate': 0.1, 'max_depth': 5, 'max_features': 3, 'min_samples_leaf': 0.2, 'min_samples_split': 0.45, 'n_estimators': 100}
dVGmimRO7bE	RF: {'bootstrap': True, 'class_weight': 0: 1, 1: 2, 'criterion': 'gini', 'max_depth': 3, 'max_features': 3, 'min_samples_leaf': 0.3, 'min_samples_split': 0.3, 'n_estimators': 20}	—	—

Usuário	C1	C2	C3
fGTU-LDm8uM	—	—	XGB: {’colsample_bytree’: .6, ’max_depth’: 15, ’n_estimators’: 20, ’reg_alpha’: 1.1, ’reg_lambda’: 1.1, ’subsample’: 0.8}
eqmPzjjzrHk	—	—	SVM: {’C’: 0.1, ’class_weight’: 0: 1, 1: 2, ’gamma’: ’scale’, ’kernel’: ’rbf’}
enJGMKaiFiI	RF: {’bootstrap’: False, ’class_weight’: ’balanced’, ’criterion’: ’entropy’, ’max_depth’: 3, ’max_features’: 1, ’min_samples_leaf’: 0.3, ’min_samples_split’: 0.3, ’n_estimators’: 30}	XGB: {’colsample_bytree’: 0.6, ’max_depth’: 15, ’n_estimators’: 20, ’reg_alpha’: 1.1, ’reg_lambda’: 1.1, ’subsample’: 0.7}	GB: {’learning_rate’: 0.1, ’max_depth’: 5, ’max_features’: 7, ’min_samples_leaf’: 0.2, ’min_samples_split’: 0.45, ’n_estimators’: 20}
fIewI06H8u8	XGB: {’colsample_bytree’: 0.6, ’max_depth’: 15, ’n_estimators’: 20, ’reg_alpha’: 1.1, ’reg$_lambda'$: 1.1, ’subsample’: 0.7}	—	—
eoWIhgawcZ0	SVM: {’C’:0.1, ’class_weight’: 0: 1, 1: 2, ’kernel’:’linear’}	—	—

Em VE EB, conforme detalhado na Tabela 4.69, os algoritmos baseados em *ensemble* e o SVM, foram encontrados como os melhores 7 vezes entre todos os cenários, com frequência 3 para SVM, 1 para RF e 3 para XGB. No caso do SVM o melhor *kernel* foi o *rbf*, com a maioria do casos com um peso igual para ambas as classes, 0 e 1. No RF o número de estimadores ficou em 20, a quantidade máxima de características por árvore de decisão ficou em 3. Para o XGB, a quantidade de estimadores ficou fixa em 20 para todos os usuários, com uma profundidade máxima para as árvores em 15.

Tabela 4.69: Hiperparâmetros algoritmos F1 a partir de 90% VE EB.

Hiperparâmetros algoritmos F1 a partir de 90% VE EB			
Usuário	**C1**	**C2**	**C3**
cFxPtyX-07w	XGB: {'colsample_bytree': 0.6, 'max_depth':15, 'n_estimators':20, 'reg_alpha':'1.1', 'reg_lambda':1.1, subsample=0.9	—	—
dyRTk2BUAeo	—	RF: {'bootstrap': True, 'class_weight': 0: 1, 1: 2, 'criterion': 'gini', 'max_depth': 3, 'max_features': 3, 'min_samples_leaf': 0.3, 'min_samples_split': 0.3, 'n_estimators': 20}	—
fIewI06H8u8	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, reg_lambda=1.1, subsample=0.8}	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'gamma': 'scale', 'kernel': 'rbf'}

Usuário	C1	C2	C3
eoWIhgawcZ0	—	SVM: {'C': 1e-07, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'}	SVM: {'C': 1e-07, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'}

Em VE EC, conforme detalhado na Tabela 4.70, para os algoritmos com hiperparâmetros, foram encontrados como os melhores 10 vezes entre todos os cenários, com frequência 2 para SVM, 2 para RF, 3 para GB e 3 para XGB. Para o SVM , quanto à análise dos hiperparâmetros, o melhor *kernel* foi o linear, sempre com a classe 1 com peso maior. Para o RF o número de estimadores variou entre 20 e 30, a quantidade máxima de características por árvore de decisão variou entre 1 e 3, com critérios entropia e *gini* respectivamente. Para o GB o número de estimadores variou entre 20 e 75, a profundidade máxima das árvores ficou em 5. Para o XGB, a quantidade de estimadores variou entre 20 e 40, com uma profundidade máxima para as árvores em 15.

Tabela 4.70: Hiperparâmetros algoritmos F1 a partir de 90% VE EC.

Hiperparâmetros algoritmos F1 a partir de 90% VE EC			
Usuário	**C1**	**C2**	**C3**
eHC7qNMAdCI	RF: {'bootstrap': False, 'class$_weight'$: 0: 1, 1: 2, 'criterion': 'gini', 'max_depth': 3, 'max_features': 5, 'min_samples_leaf': 0.3, 'min_samples_split': 0.3, 'n_estimators': 20}	—	—
dyRTk2BUAeo	—	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 40, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	—

Usuário	C1	C2	C3
cFxPtyX-07w	SVM: {'C': 0.001, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear' }	—	—
fGTU-LDm8uM	—	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	RF: {'bootstrap': False, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 3, 'max_features': 3, 'min_samples_leaf': 0.3, 'min_samples_split': 0.3, 'n_estimators': 20}
enJGMKaiFiI	GB: {'learning_rate': 0.001, 'max_depth': 5, 'max_features': 3, 'min_samples_leaf': 0.2, 'min_samples_split': 0.3, 'n_estimators': 30}	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	GB: {'learning_rate': 0.001, 'max_depth': 5, 'max_features': 7, 'min_samples_leaf': 0.2, 'min$_s amples_split'$: 0.3, 'n_estimators': 75}
fIewI06H8u8	—	—	GB: {'learning_rate': 0.01, 'max_depth': 5, 'max_features': 7, 'min_samples_leaf': 0.2, 'min_samples_split': 0.3, 'n_estimators': 20}

Usuário	C1	C2	C3
eoWIhgawcZ0	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear'}	—	—

Em VE ED, conforme detalhado na Tabela 4.71, o algoritmo RF foi encontrado como o melhor para todos os cenários. Com um valor de estimadores variando entre 40 e 50, o critério foi sempre entropia e a quantidade máxima de características a ser sorteada por árvore variou entre 9 e 11.

Tabela 4.71: Hiperparâmetros algoritmos F1 a partir de 90% VE ED.

Hiperparâmetros algoritmos F1 a partir de 90% VE ED			
Usuário	**C1**	**C2**	**C3**
Todos	RF: {'n_estimators': 50, 'n_jobs': 2, 'random_state':0, 'bootstrap': 'False', 'criterion': 'entropy', 'max_depth': 7, 'max_features': 11, 'min_samples_leaf': 6}	RF: {'n_estimators': 50, 'n_jobs': 2, 'random_state':0, 'bootstrap': 'False', 'criterion': 'entropy', 'max_depth': 7, 'max_features': 11, 'min_samples_leaf': 6}	RF: {'bootstrap': True, 'class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': None, 'max_features': 9, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 40}

Em VE EE, conforme detalhado na Tabela 4.72, o algoritmo XGB foi encontrado como o melhor para os cenários 1 e 2 e para o cenário 3 o melhor foi o RF. Para o XGB, a quantidade de estimadores variou entre 50 e 100, a profundidade da árvore foi mantida em 15 tanto para os cenários 1 e 2. Para o RF o número de estimadores foi de 30, com a quantidade máxima de características por árvore em 9 e com critério *gini*.

Tabela 4.72: Hiperparâmetros algoritmos F1 a partir de 90% VE EE.

Hiperparâmetros algoritmos F1 a partir de 90% VE EE			
Usuário	**C1**	**C2**	**C3**
Todos	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 50, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.6}	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 100, 'reg_alpha': 1.3, 'reg_lambda': 1.3, 'subsample': 0.7}	RF: {'n_estimators': 30, 'n_jobs': 2, 'random_state': 0, 'bootstrap': 'True', 'class_weight': 'balanced', 'criterion': 'gini', 'max_depth': None, 'max_features': 9, 'min_samples_leaf': 1, 'min_samples_split': 6}

Em VE EF, conforme detalhado na Tabela 4.73, o algoritmo RF foi o melhor no cenário 1 e o XGB foi encontrado como o melhor para os cenários 2 e 3. Para o RF o número de estimadores foi de 30, com a quantidade máxima de características por árvore em 5 e com critério *gini*. Para o XGB, a quantidade de estimadores se manteve fixa em 100, a profundidade da árvore foi também mantida em 15 tanto para os cenários 2 e 3.

Tabela 4.73: Hiperparâmetros algoritmos F1 apartir de 90% VE EF.

Hiperparâmetros algoritmos F1 a partir de 90% VE EF			
Usuário	**C1**	**C2**	**C3**
Todos	RF: {'n_estimators': 25, 'n_jobs': 2, random_state=0, 'bootstrap': True, 'criterion': 'gini', 'max_depth': None, 'max_features': 5, 'min_samples_leaf':5 min_samples_split: 2, class_weight; 'balanced' }	XGB: {colsample_bytree: 0.6, max_depth: 15, n_estimators: 100, reg_alpha; 1.3, reg_lambda: 1.3, subsample: 0.7}	XGB: {colsample_bytree: 0.6, max_depth: 15, n_estimators: 100, reg_alpha: 1.3, reg_lambda: 1.3, subsample: 0.7}

Em VD EA, conforme detalhado na Tabela 4.74, os algoritmos baseados em *ensemble* e o SVM, foram encontrados como os melhores 11 vezes entre todos os cenários, com frequência 5 para SVM, 1 para RF, 2 para GB e 3 para XGB. No caso do SVM, quanto à análise dos hiperparâmetros, o *kernel* predominante foi o linear, sempre com peso 2 para a classe 1. Para o RF o número de estimadores foi 25, a quantidade máxima de características por árvore de decisão ficou em 3. Para o GB a quantidade de estimadores foi 20. Para o XGB o número de estimadores variou entre 20 e 30, a profundidade máxima para as árvores ficou fixa em 15 para todos os casos.

Tabela 4.74: Hiperparâmetros algoritmos F1 a partir de 90% VD EA.

Hiperparâmetros algoritmos F1 a partir de 90% VD EA			
Usuário	**C1**	**C2**	**C3**
fUB30EtiU0Q	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	—	—
dyRTk2BUAeo	—	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'gamma': 'auto', 'kernel': 'rbf'}	—
fmVXDwdw20Q	—	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear'}	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear'}
eqmPzjjzrHk	—	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear'}	—

Usuário	C1	C2	C3
enJGMKaiFiI	—	XGB: {’colsample_bytree’: 0.6, ’max_depth’: 15, ’n_estimators’: 20, ’reg_alpha’: 1.1, ’reg_lambda’: 1.1, ’subsample’: 0.7}	XGB: {’colsample_bytree’: 0.6, ’max_depth’: 15, ’n_estimators’: 30, ’reg_alpha’: 1.1, ’reg_lambda’: 1.1, ’subsample’: 0.7}
dlj7Igoq3HQ	—	GB: {’learning_rate’: 0.001, ’max_depth’: 5, ’max_features’: 3, ’min_samples_leaf’: 0.2, ’min_samples_split’: 0.3, ’n_estimators’: 20}	SVM: {’C’: 0.1, ’class_weight’: 0: 1, 1: 2, ’kernel’: ’linear’}
fhw9jzhvkGs	—	RF: {’bootstrap’: True, ’class_weight’: ’balanced’, ’criterion’: ’gini’, ’max_depth’: 3, ’max_features’: 3, ’min_samples_leaf’: 0.3, ’min_samples_split’: 0.3, ’n_estimators’: 25}	GB: {’learning_rate’: 0.001, ’max_depth’: 5, ’max_features’: 3, ’min_samples_leaf’: 0.2, ’min_samples_split’: 0.3, ’n_estimators’: 20}

Em VD EB, conforme detalhado na Tabela 4.75, os algoritmos baseados em *ensemble* e o SVM, foram encontrados como os melhores com frequência 1 para SVM e 4 para XGB. Para o SVM, quanto à análise dos hiperparâmetros, o melhor *kernel* foi o rbf, com o mesmo peso para as classes 0 e 1. Para o XGB o número de estimadores variou entre 20 e 100, a profundidade máxima para as árvores ficou fixa em 15 para todos os casos.

Tabela 4.75: Hiperparâmetros algoritmos F1 a partir de 90%VD EB.

Hiperparâmetros algoritmos F1 a partir de 90% VD EB			
Usuário	**C1**	**C2**	**C3**
cBc3b9Cv4X0	XGB: {'colsample_bytree': 0.8, 'max_depth': 15, 'n_estimators': 30, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.8}	—	—
fmVXDwdw20Q	—	—	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}
enJGMKaiFiI	—	—	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}
eoWIhgawcZ0	SVM: {'C': 1e-07, 'class_weight': 'balanced', 'gamma': 0.001, 'kernel': 'rbf'}	—	—
fhw9jzhvkGs	—	—	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 100, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}

Em VD EC, conforme detalhado na Tabela 4.76, os algoritmos baseados em *ensemble* e o SVM, foram encontrados como os melhores 8 vezes entre todos os cenários, com frequência 1 para SVM, 2 para RF, 3 para GB e 2 para XGB. No caso do SVM, quanto à análise dos hiperparâmetros, o melhor *kernel* foi o linear, com peso 2 para a classe 1. Para o RF o número de estimadores variou entre 20 e 25, a quantidade máxima de características por árvore de decisão ficou em 3. Para o GB a quantidade de estimadores variou entre 20 e 100. Para o XGB o número de estimadores ficou em 20, a profundidade máxima para as árvores ficou fixa em 15 para todos os casos.

Tabela 4.76: Hiperparâmetros algoritmos F1 a partir de 90% VD EC.

Hiperparâmetros algoritmos F1 a partir de 90% VD EC			
Usuário	**C1**	**C2**	**C3**
fUB30EtiU0Q	XGB: {'colsample_bytree': 0.6, 'max_depth': 15, 'n_estimators': 20, 'reg_alpha': 1.1, 'reg_lambda': 1.1, 'subsample': 0.7}	—	—
fmVXDwdw20Q	—	SVM: {'C': 0.1, 'class_weight': 0: 1, 1: 2, 'kernel': 'linear'}	RF: {'bootstrap': False, 'class$_weight'$: 0: 1, 1: 2, 'criterion': 'gini', 'max_depth': 3, 'max_features': 3, 'min_samples_leaf': 0.3, 'min_samples_split': 0.3, 'n_estimators': 25}

Usuário	C1	C2	C3
eqmPzjjzrHk	RF: {’bootstrap’: True, ’class_weight’: ’balanced’, ’criterion’: ’gini’, ’max_depth’: 3, ’max_features’: 3, ’min_samples_leaf’: 0.3, ’min_samples_split’: 0.3, ’n_estimators’: 20}	—	—
enJGMKaiFiI	—	GB: {’learning_rate’: 0.001, ’max_depth’: 5, ’max_features’: 3, ’min_samples_leaf’: 0.4, ’min_samples_split’: 0.3, ’n_estimators’: 100}	XGB: {’colsample_bytree’: 0.6, ’max_depth’: 15, ’n_estimators’: 20, ’reg_alpha’: 1.1, ’reg_lambda’: 1.1, ’subsample’: 0.7}
dlj7Igoq3HQ	—	GB: {’learning_rate’: 0.001, ’max_depth’: 5, ’max_features’: 3, ’min_samples_leaf’: 0.2, ’min_samples_split’: 0.3, ’n_estimators’: 20}	—

Usuário	C1	C2	C3
fhw9jzhvkGs	—	—	GB: {’learning_rate’: 0.001, ’max_depth’: 5, ’max_features’: 3, ’min_samples_leaf’: 0.2, ’min_samples_split’: 0.3, ’n_estimators’: 75}

Em VD ED, conforme detalhado na Tabela 4.77, o algoritmo RF foi encontrado como os melhores para os cenários 1 e 2. Com um valor de estimadores variando entre 20 e 40, para o cenário 1, o critério foi *gini* e para o cenário 2 entropia, a quantidade máxima de características a ser sorteada por árvore variou entre 1 e 9.

Tabela 4.77: Hiperparâmetros algoritmos F1 a partir de 90%VD ED.

Hiperparâmetros algoritmos F1 a partir de 90%VD ED			
Usuário	**C1**	**C2**	**C3**
Todos	RF: {’n_estimators’: 20, ’n_jobs’: 2, random_state=0, ’bootstrap’: True, ’criterion’: ’gini’, ’max_depth’: 3, ’max_features’: 1, ’min_samples_leaf’:1}	RF: {’bootstrap’: False, ’class_weight’: ’balanced’, ’criterion’: ’entropy’, ’max_depth’: None, ’max_features’: 9, ’min_samples_leaf’: 1, ’min_samples_split’: 2, ’n_estimators’: 40}	—

Em VD EE, conforme detalhado na Tabela 4.78, um algoritmo baseado em *ensemble* foi encontrado como o melhor apenas no cenário 3 com o GB. Com um valor de estimador em 100 e a quantidade máxima de características a ser sorteada por árvore foi definida em 10.

Tabela 4.78: Hiperparâmetros algoritmos F1 a partir de 90%VD EE.

Hiperparâmetros algoritmos F1 a partir de 90%VD EE			
Usuário	**C1**	**C2**	**C3**
Todos	—	—	GB: {'learning_rate': 0.1, 'max_depth': 5, 'max_features': 10, 'min_samples_leaf': 0.2, 'min_samples_split': 0.3, 'n_estimators': 100, 'random_state': 0}

Em VD EF, conforme detalhado na Tabela 4.79, um algoritmo com hiperparâmetros foi encontrado como o melhor apenas no cenário 3 com o SVM. Com *kernel* linear e peso das classes balanceado.

Tabela 4.79: Hiperparâmetros algoritmos F1 a partir de 90% VD EF.

Hiperparâmetros algoritmos F1 a partir de 90% VD EF			
Usuário	**C1**	**C2**	**C3**
Todos	—	SVM: {C: 0.1, class_weight: 'balanced', degree: 5, kernel: 'linear', gamma: 1e-2}	—

A análise dos hiperparâmetros para os melhores algoritmos baseados em *ensemble* e o SVM, entre todos os cenários demonstram que esses parâmetros podem variar bastante entre os usuários. Isso indica como é complexa a tarefa de trabalhar com algoritmos que podem receber hiperparâmetros, pois existe um grande potencial neles para a resolução de problemas, mas também existe o desafio de encontrar os melhores hiperparâmetros que ajustam o algoritmo da melhor forma para o problema, sem *overfitting* ou *underfitting*.

Os resultados para os melhores hiperparâmetros também indicam como os dados coletados de usuários, interagindo com uma mesma aplicação podem produzir coleções complexas que tornam difícil encontrar um único algoritmo que consiga ser o melhor em F1 em todos os casos, reforçando a importância da metodologia abordada no *framework* proposto, de utilizar uma gama de 6 algoritmos de Aprendizado de Máquina diferentes para encontrar a melhor performance por usuário.

4.4 COMPARAÇÃO CARACTERÍSTICAS ENTRE OS USUÁRIOS

Para evidênciar a diferença entre os valores de uma mesma característica entre os usuários, foi capturado o valor máximo e mínimo de cada uma das características coletadas nos Momentos 1 e 2, e esses serão detalhados em gráficos nesta seção.

4.4.1 Comparação Verificação Estática

4.4.1.1 *Touchscreen*

Os valores de tempo pressiona pressiona tiveram uma variação mais acentuada para três usuários, os outros apresentaram valores próximos, e em muitos casos aparentemente iguais, conforme demonstrado na Figura 4.1. Para dois usuários estão bem maiores em relação aos outros podendo indicar a presença de *outliers* nos dados destes usuários, ou apenas são usuários com um comportamento diferente da média.

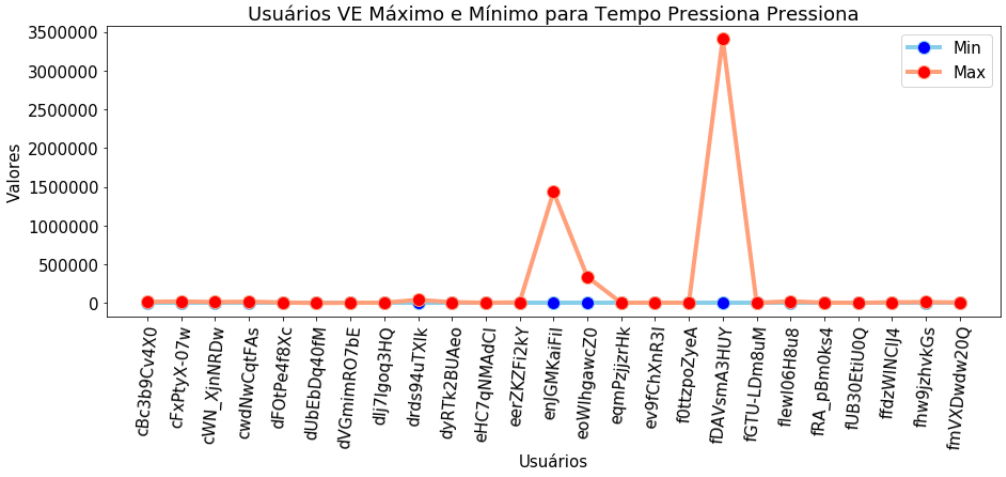

Figura 4.1: Gráfico VE valores máximos e mínimos para tempo pressiona pressiona.

Os valores de tempo pressiona solta tiveram uma alta variação entre os usuários, conforme demonstrado na Figura 4.2.

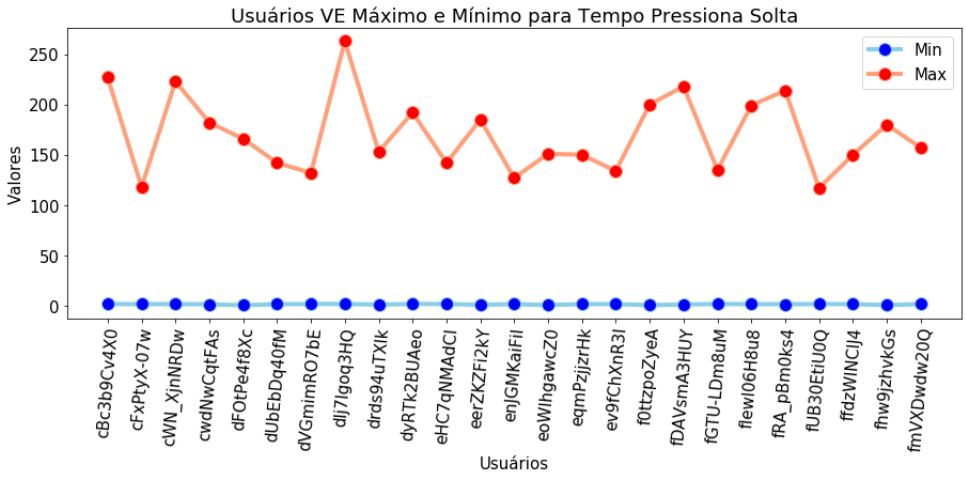

Figura 4.2: Gráfico VE valores máximos e mínimos para tempo pressiona solta.

Os valores de tempo solta pressiona tiveram no geral uma baixa variação entre os usuários, e formou um gráfico bem semelhante ao do tempo pressiona pressiona, conforme demonstrado na Figura 4.3.

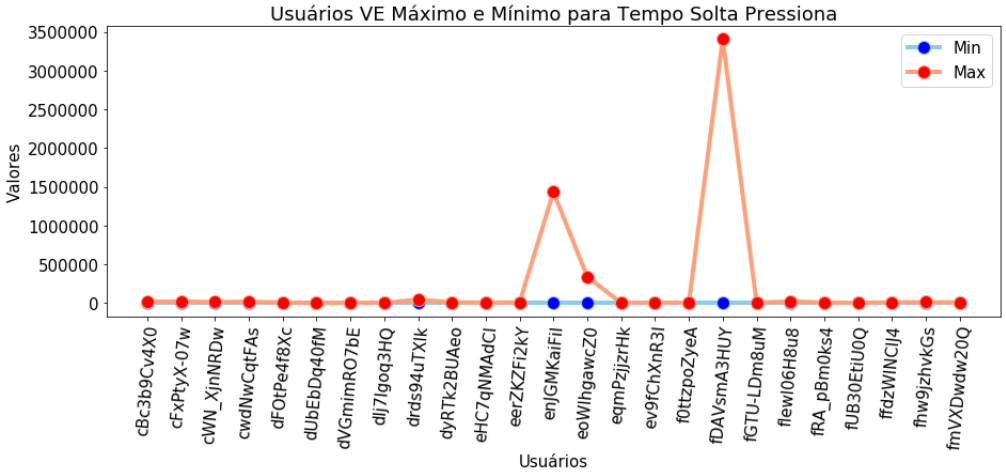

Figura 4.3: Gráfico VE valores máximos e mínimos para tempo solta pressiona.

Os valores de tempo solta solta tiveram no geral uma baixa variação entre os usuários, e formou um gráfico bem semelhante ao do tempo pressiona pressiona e ao do solta pressiona, fazendo com que se chegue à conclusão que essa característica podem não ser tão relevante para a criação dos modelos, conforme demonstrado na Figura 4.4.

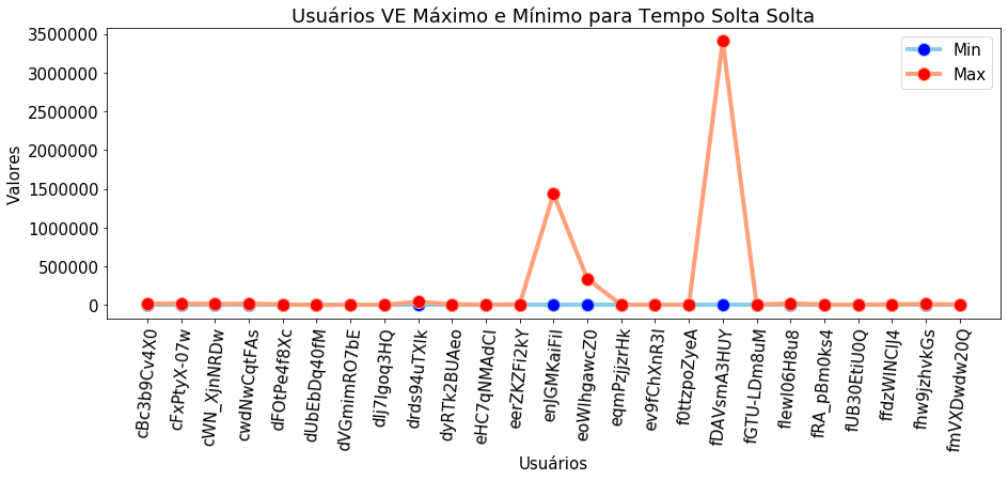

Figura 4.4: Gráfico VE valores máximos e mínimos para tempo solta solta.

Os valores de média do tempo de pressionamento teve uma alta variação entre os usuários, demonstrando que cada usuário teve um padrão diferente em relação à esta característica, conforme demonstrado na Figura 4.5.

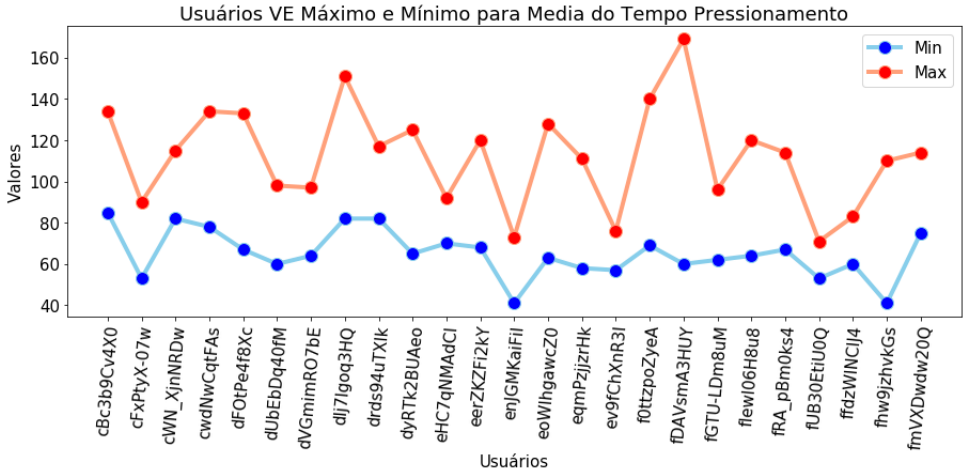

Figura 4.5: Gráfico VE valores máximos e mínimos para média do tempo de pressionamento.

Os valores de pressão para VE teve uma variação maior para 7 usuários, os outros usuários apresentaram valores máximos e mínimos próximos, conforme demonstrado na Figura 4.6.

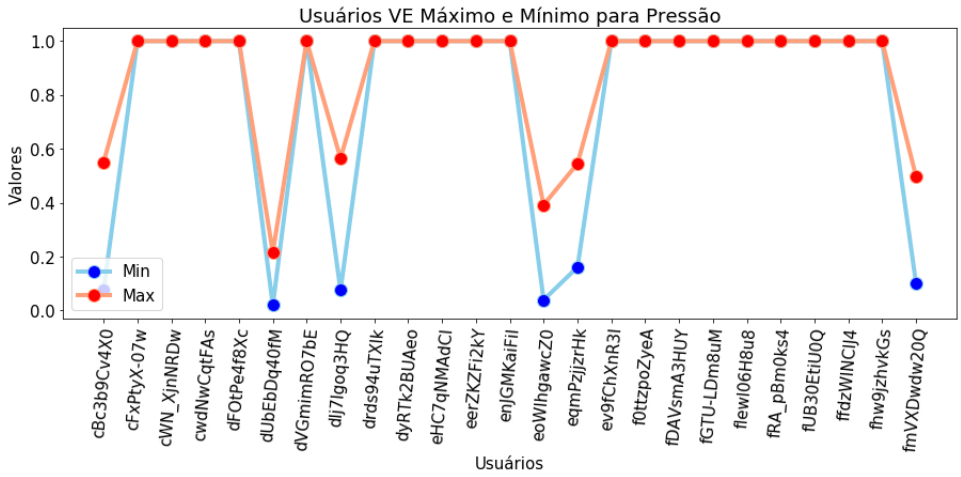

Figura 4.6: Gráfico VE valores máximos e mínimos para pressão.

Como os valores de pressão tiveram uma alta variação entre os valores máximos e mínimos dos usuários, fazendo com que não seja fácil visualizar os valores na Figura 4.6, com o objetivo de deixar mais clara a variação dos valores entre os usuários foi criada a Tabela 4.80, onde pode ser observado que para a grande maioria dos usuários o valor retornado é 1, indicando que o sensor apenas consegue retornar um valor único.

Tabela 4.80: Valores máximos e mínimos VE pressão.

Usuário	**Presão**	
Identificação	**Valor Mínimo**	**Valor Máximo**
drds94uTXlk	1,000000	1,000000
cwdNwCqtFAs	1,000000	1,000000
eHC7qNMAdCI	1,000000	1,000000
dUbEbDq40fM	0,019608	0,215686
fUB30EtiU0Q	1,000000	1,000000
cWN_XjnNRDw	1,000000	1,000000
ffdzWINClJ4	1,000000	1,000000
dFOtPe4f8Xc	1,000000	1,000000
cBc3b9Cv4X0	0,075000	0,550000
ev9fChXnR3I	1,000000	1,000000
fRA_pBm0ks4	1,000000	1,000000
dyRTk2BUAeo	1,000000	1,000000
cFxPtyX-07w	1,000000	1,000000
dVGmimRO7bE	1,000000	1,000000
fGTU-LDm8uM	1,000000	1,000000
fmVXDwdw20Q	0,101961	0,498039
eqmPzjjzrHk	0,160784	0,545098
enJGMKaiFiI	1,000000	1,000000

Identificação	Valor Mínimo	Valor Máximo
fIewI06H8u8	1,000000	1,000000
eerZKZFi2kY	1,000000	1,000000
dlj7Igoq3HQ	0,075000	0,565000
fDAVsmA3HUY	1,000000	1,000000
f0ttzpoZyeA	1,000000	1,000000
eoWIhgawcZ0	0,039216	0,392157
fhw9jzhvkGs	1,000000	1,000000

Os valores de média da pressão para VE apresentaram uma variação maior em relação a pressão, demonstrando que os usuários aplicam em média uma pressão diferente na interação com uma mesma aplicação mobile, mas para alguns usuários os valores ainda foram próximos, conforme demonstrado na Figura 4.7.

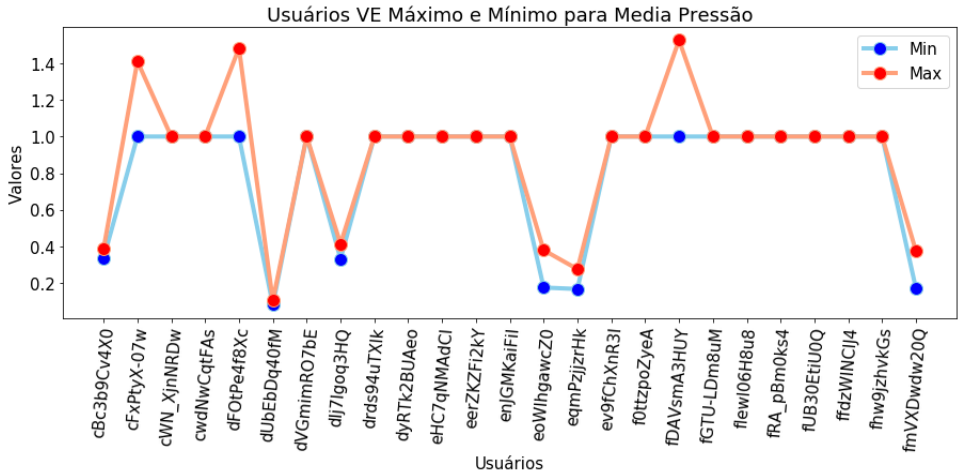

Figura 4.7: Gráfico VE valores máximos e mínimos para média pressão.

Os valores de tamanho do dedo para VE apresentaram uma variação entre os usuários, para dois usuários os valores de máximo ficaram bem acima dos demais, o que prejudicou a melhor visualização da variação dos valores de máximo e mínimo dos outros 23 usuários, conforme demonstrado na Figura 4.8.

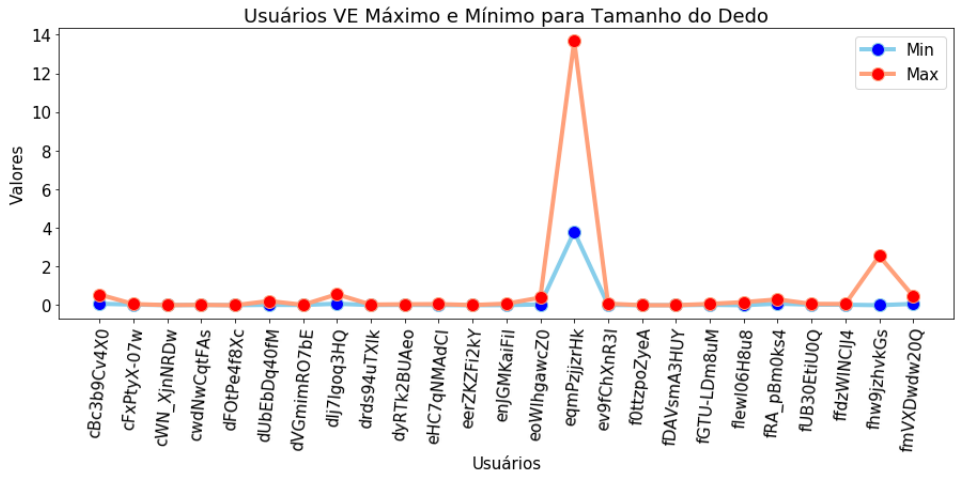

Figura 4.8: Gráfico VE valores máximos e mínimos para tamanho do dedo.

Como os valores do tamanho do dedo tiveram uma variação de valores máximos e mínimos muito grandes entre o usuários, impossibilitando a correta visualização da diferença para estes valores entre estes na Figura 4.8, foi criada a Tabela 4.81 para detalhar a diferença de valores máximos e mínimos entre os usuários de forma mais clara, o que deixou evidente que os valores são realmente bem distintos entre os usuários.

Tabela 4.81: Valores máximos e mínimos VE tamanho do dedo.

Usuário	**Tamanho do dedo**	
Identificação	**Valor Mínimo**	**Valor Máximo**
drds94uTXlk	0,007843	0,031373
cwdNwCqtFAs	0,011765	0,015686
eHC7qNMAdCI	0,007843	0,054902
dUbEbDq40fM	0,019608	0,215686
fUB30EtiU0Q	0,027451	0,066667
cWN_XjnNRDw	0,003922	0,003922
ffdzWINClJ4	0,023529	0,062745
dFOtPe4f8Xc	0,000000	0,000000
cBc3b9Cv4X0	0,075000	0,550000
ev9fChXnR3I	0,015686	0,078431
fRA_pBm0ks4	0,080000	0,300000
dyRTk2BUAeo	0,015686	0,047059
cFxPtyX-07w	0,031373	0,058824
dVGmimRO7bE	0,009804	0,019608
fGTU-LDm8uM	0,019608	0,066667
fmVXDwdw20Q	0,066667	0,466667
eqmPzjjzrHk	3,800000	13,666667
enJGMKaiFiI	0,003922	0,078431

Identificação	Valor Mínimo	Valor Máximo
fIewI06H8u8	0,000000	0,158824
eerZKZFi2kY	0,000000	0,007874
dlj7Igoq3HQ	0,075000	0,565000
fDAVsmA3HUY	0,000000	0,000000
f0ttzpoZyeA	0,000000	0,000000
eoWIhgawcZ0	0,039216	0,392157
fhw9jzhvkGs	0,000000	2,550000

Os valores de média do tamanho do dedo para VE apresentaram uma variação entre os usuários, assim como aconteceu para o tamanho do dedo, dois usuários tiveram valores de máximo muito acima dos demais, o que prejudicou, novamente, a melhor visualização da variação dos valores de máximo e mínimo dos outros 23 usuários. Indicando que estes usuários tem um tamanho de área de contato do dedo com a tela bem maior que os outros, conforme demonstrado na Figura 4.9.

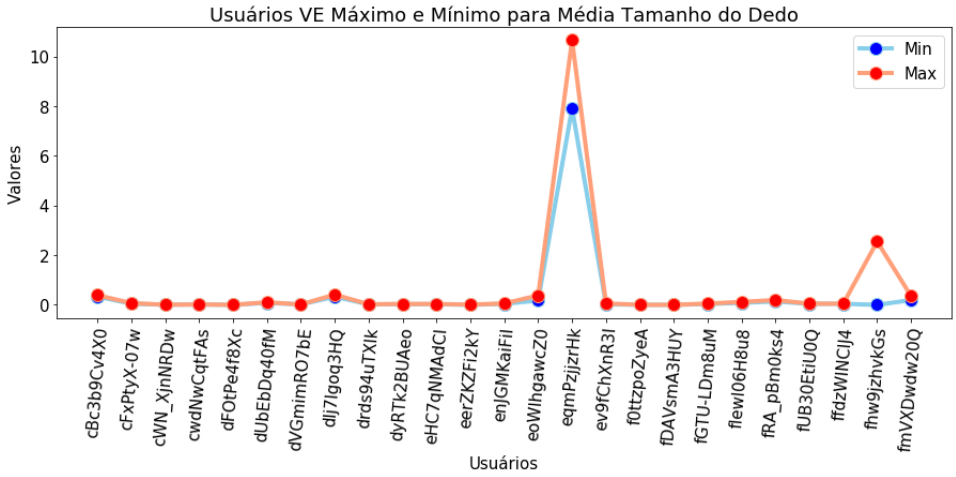

Figura 4.9: Gráfico VE valores máximos e mínimos para média do tamanho do dedo.

4.4.1.2 Acelerômetro

Os valores de aceleração ao longo do eixo X, Y e Z em VE, geradas pelo acelerômetro, tiveram uma alta variação entre os usuários, indicando que estas podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.10, 4.11 e 4.12.

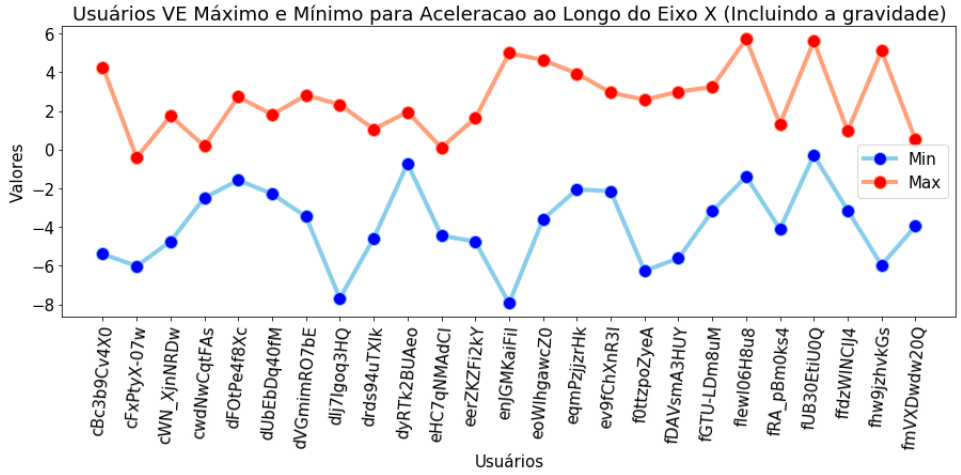

Figura 4.10: Gráfico VE valores máximos e mínimos para aceleração ao longo do eixo X (incluindo a gravidade).

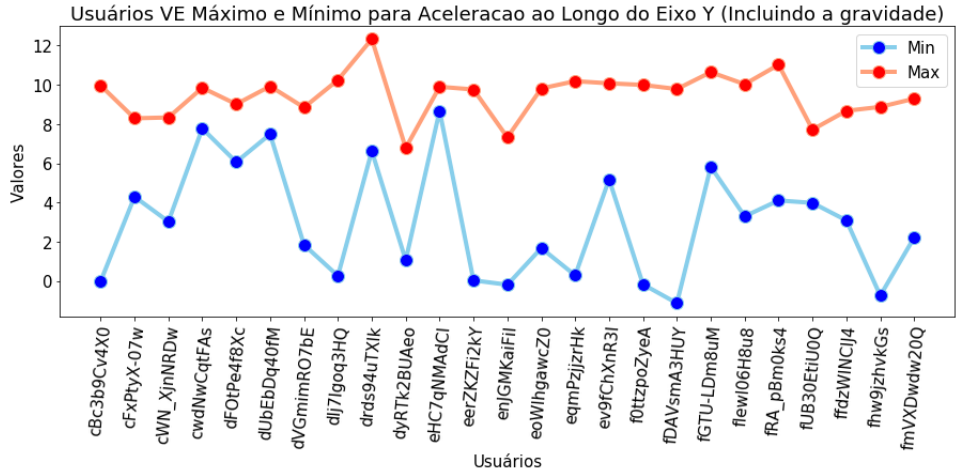

Figura 4.11: Gráfico VE valores máximos e mínimos para aceleração ao longo do eixo Y (incluindo a gravidade).

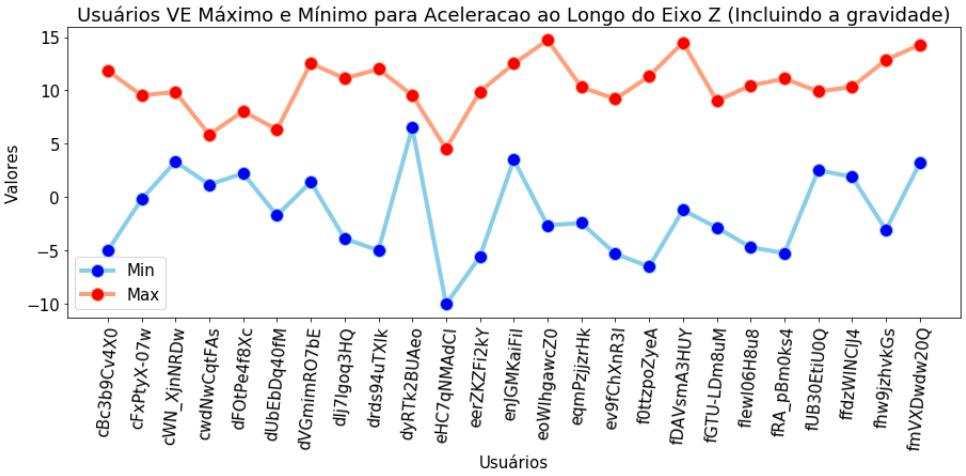

Figura 4.12: Gráfico VE valores máximos e mínimos para aceleração ao longo do eixo Z (incluindo a gravidade).

4.4.1.3 Giroscópio

Os valores de rotação ao redor de X, Y e Z em VE, geradas pelo giroscópio, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.13, 4.14, e 4.15.

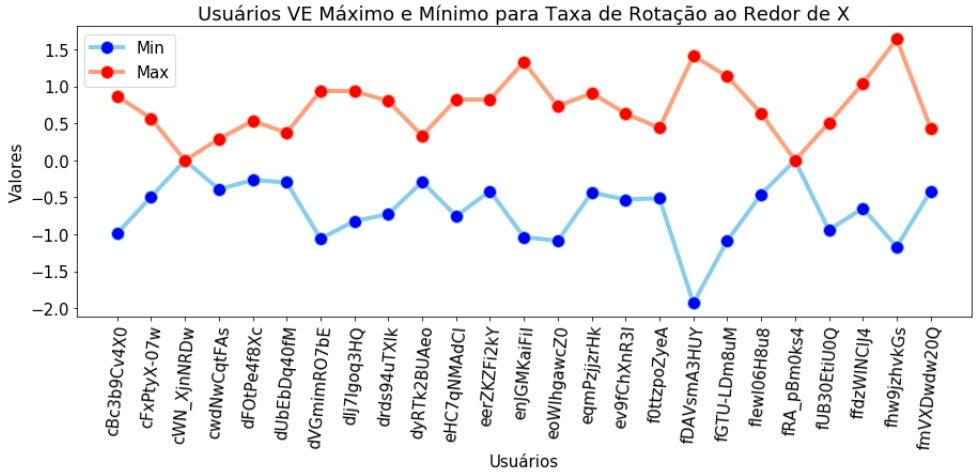

Figura 4.13: Gráfico VE valores máximos e mínimos para taxa de rotação ao redor de X.

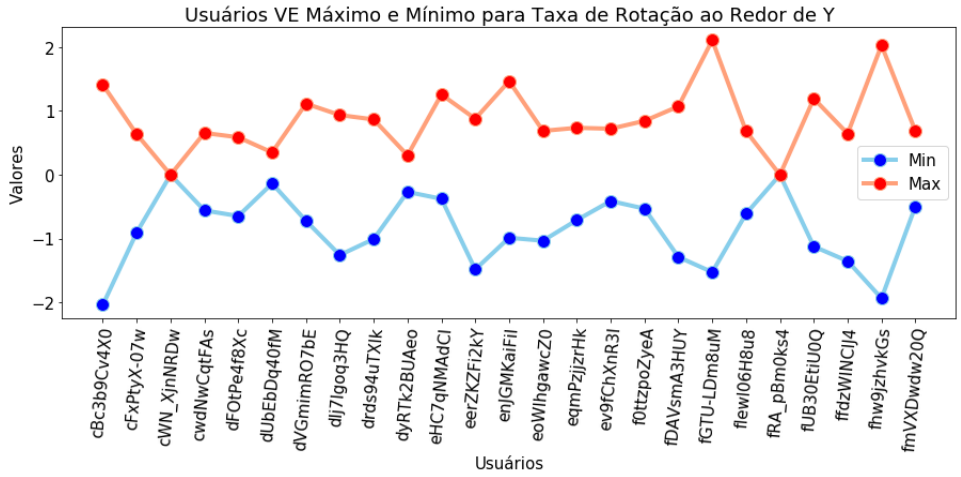

Figura 4.14: Gráfico VE valores máximos e mínimos para taxa de rotação ao redor de Y.

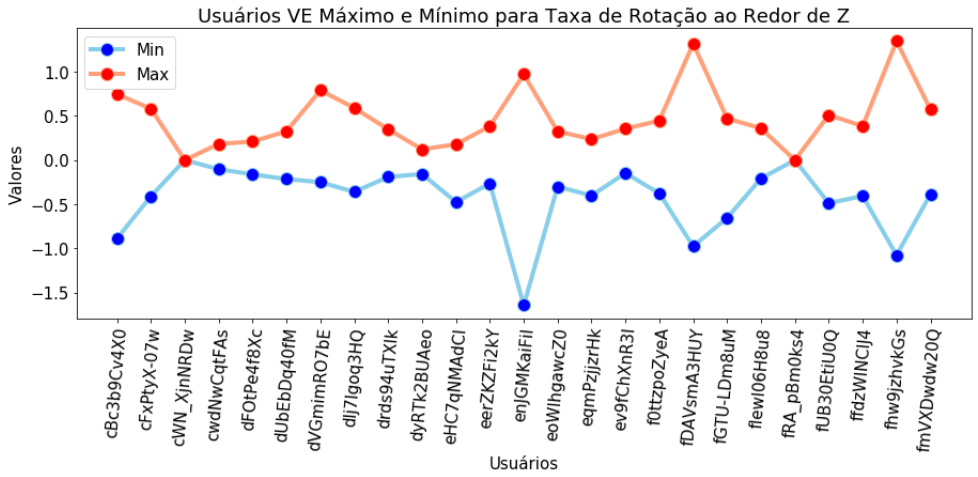

Figura 4.15: Gráfico VE valores máximos e mínimos para taxa de rotação ao redor de Z.

4.4.1.4 Magnetômetro

Os valores para o campo geomagnético sobre X, Y e Z em VE, geradas pelo magnetômetro, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.16, 4.17, e 4.18.

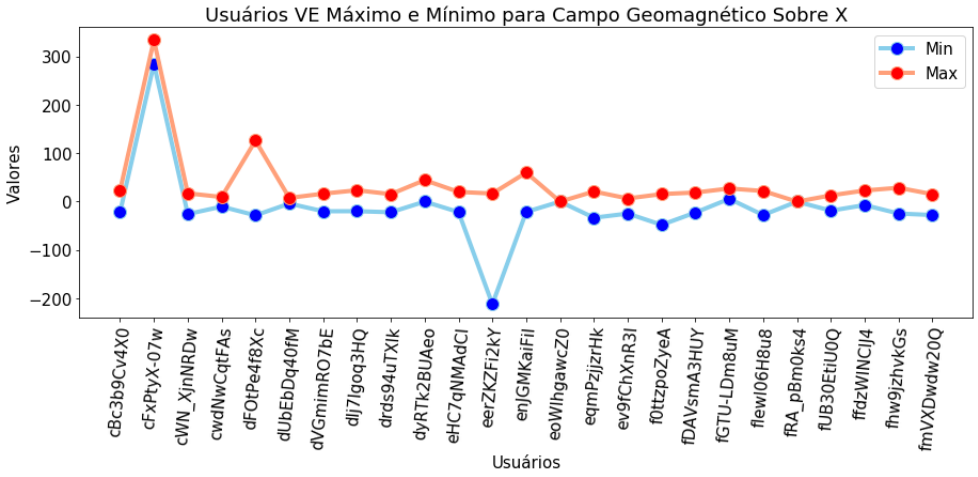

Figura 4.16: Gráfico VE valores máximos e mínimos para campo geomagnético sobre X.

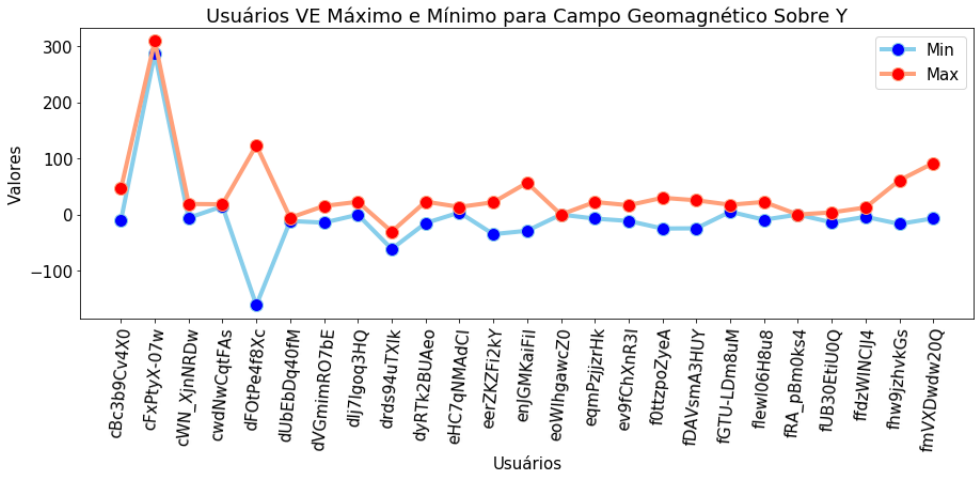

Figura 4.17: Gráfico VE valores máximos e mínimos para campo geomagnético sobre Y.

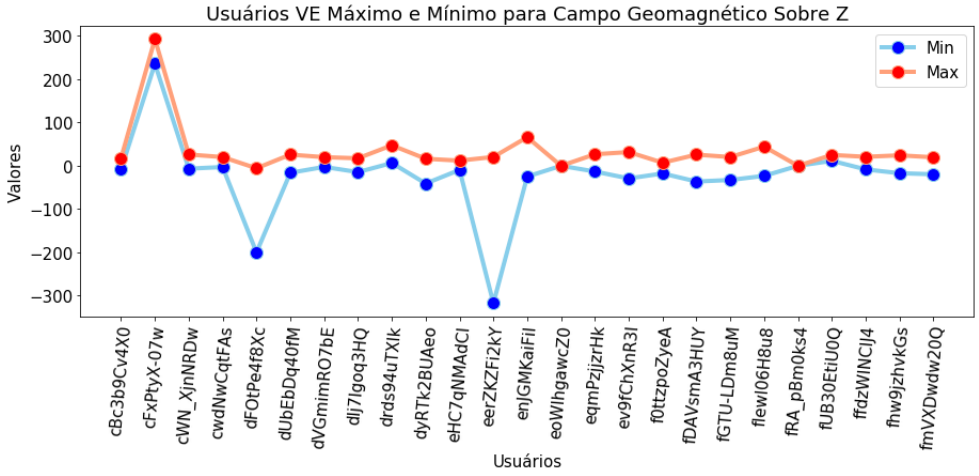

Figura 4.18: Gráfico VE valores máximos e mínimos para campo geomagnético sobre Z.

4.4.1.5 Sensores de Rotação

No caso das características geradas a partir dos sensores de rotação em VE, a acurácia estimada foi a que apresentou um grau menor de variação para máximos e mínimos entre os usuários. Já os valores para o componente escalar vetor rotação, e os componente do vetor de rotação ao longo de X, Y e Z, tiveram uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.19, 4.20, 4.21, 4.22, e 4.23.

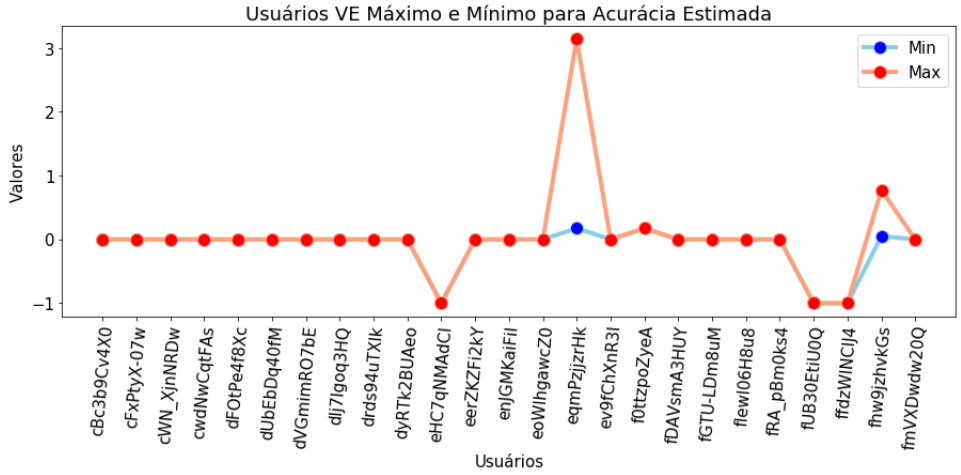

Figura 4.19: Gráfico VE valores máximos e mínimos para acurácia estimada.

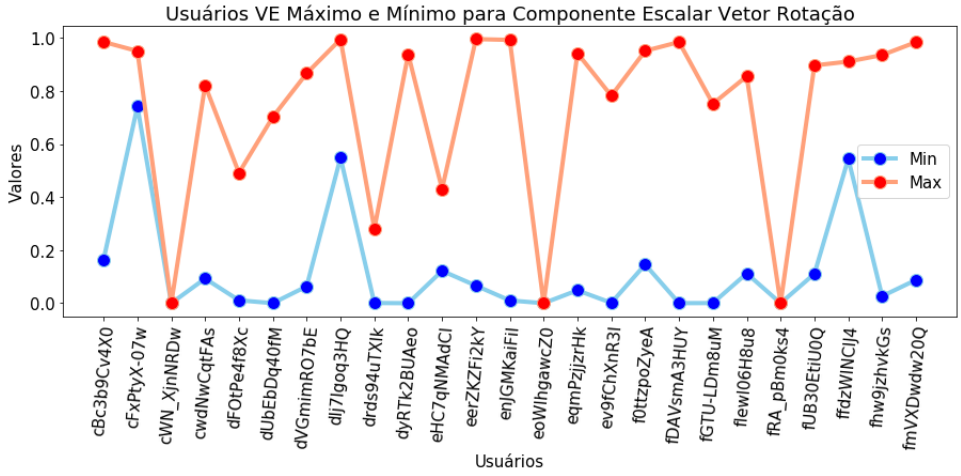

Figura 4.20: Gráfico VE valores máximos e mínimos para componente escalar vetor rotação.

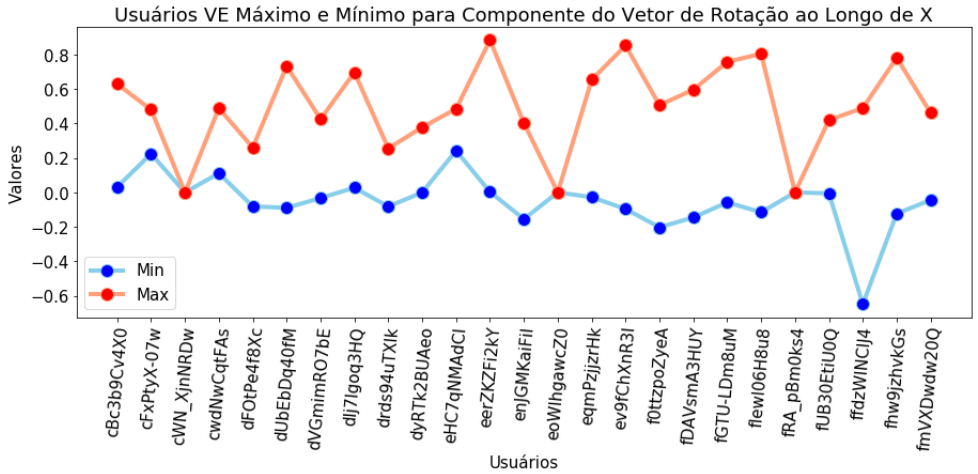

Figura 4.21: Gráfico VE valores máximos e mínimos para componente do vetor de rotação ao longo de X.

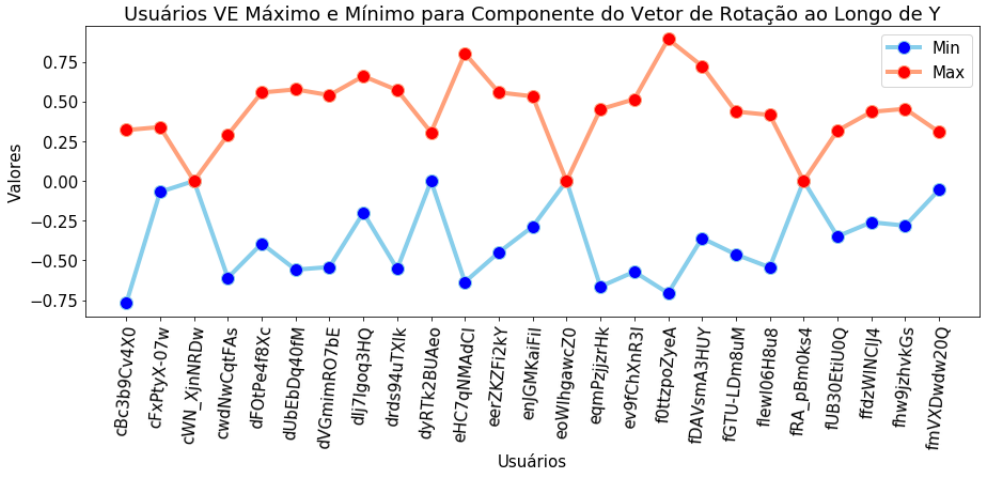

Figura 4.22: Gráfico VE valores máximos e mínimos para componente do vetor de rotação ao longo de Y.

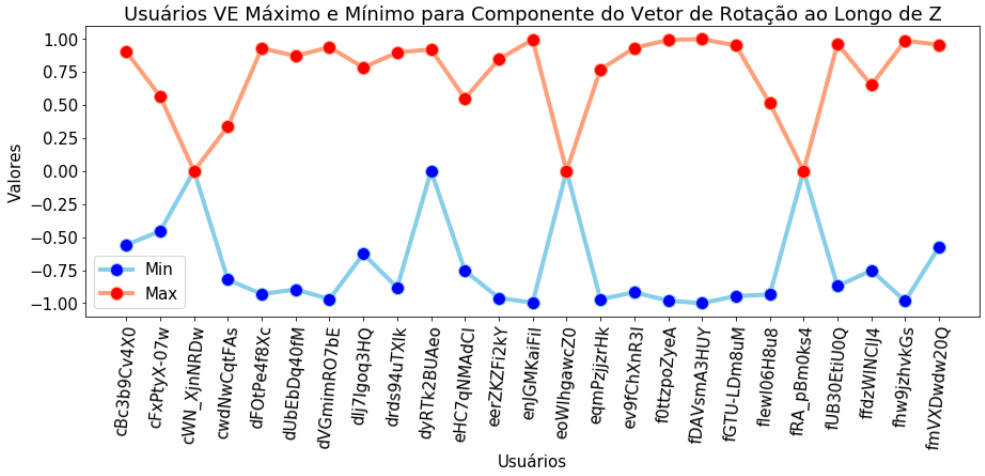

Figura 4.23: Gráfico VE valores máximos e mínimos para componente do vetor de rotação ao longo de Z.

4.4.1.6 Sensores de Aceleração

Os valores para força da aceleração longo de X, Y e Z em VE, excluindo a gravidade, geradas pelos sensores de aceleração, variou também entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.24, 4.25, e 4.26.

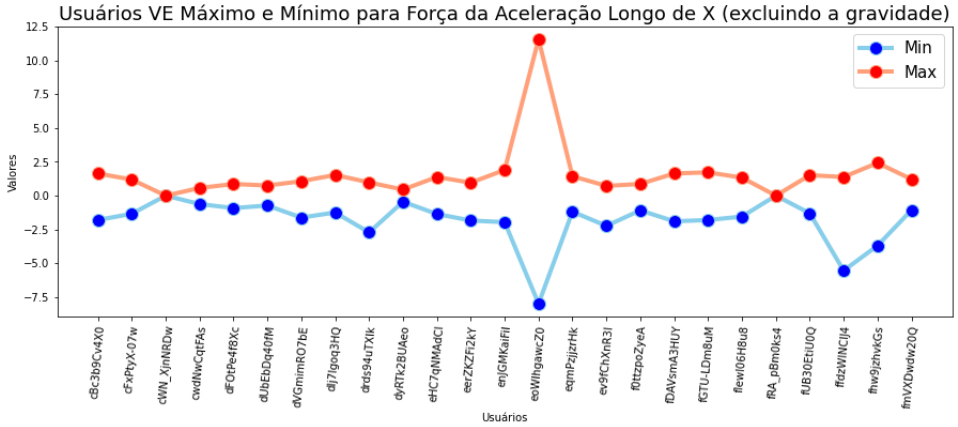

Figura 4.24: Gráfico VE valores máximos e mínimos para força da aceleração longo de X (excluindo a gravidade).

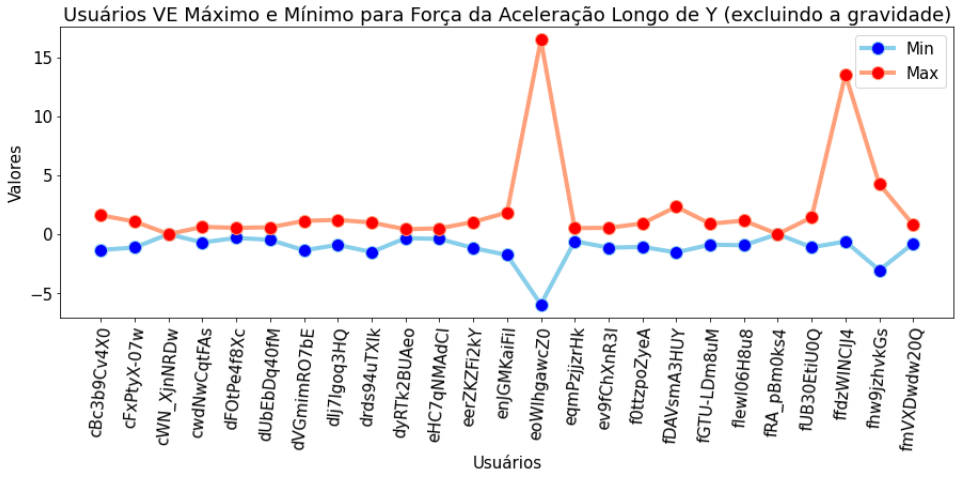

Figura 4.25: Gráfico VE valores máximos e mínimos para força da aceleração longo de Y (excluindo a gravidade).

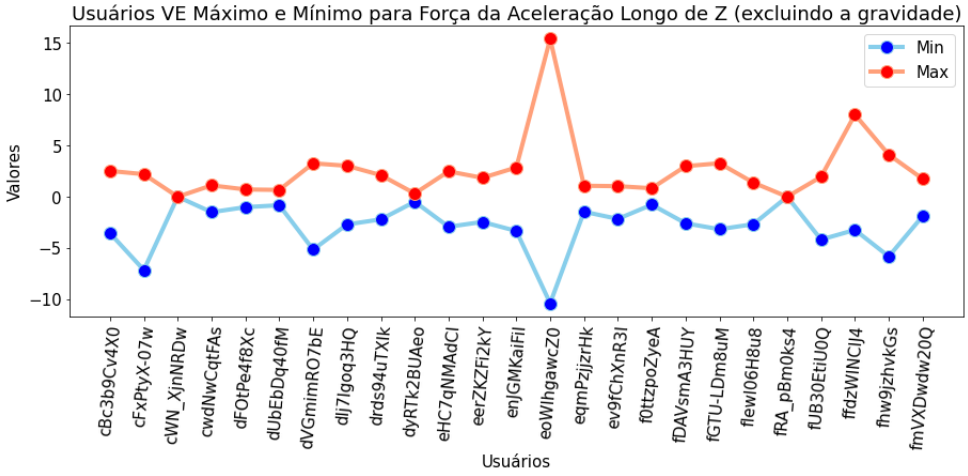

Figura 4.26: Gráfico VE valores máximos e mínimos para força da aceleração longo de Z (excluindo a gravidade).

4.4.1.7 Sensores de Gravidade

Os valores para força da gravidade ao longo de X, Y e Z em VE, geradas pelos sensores de gravidade, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.27, 4.28, e 4.29.

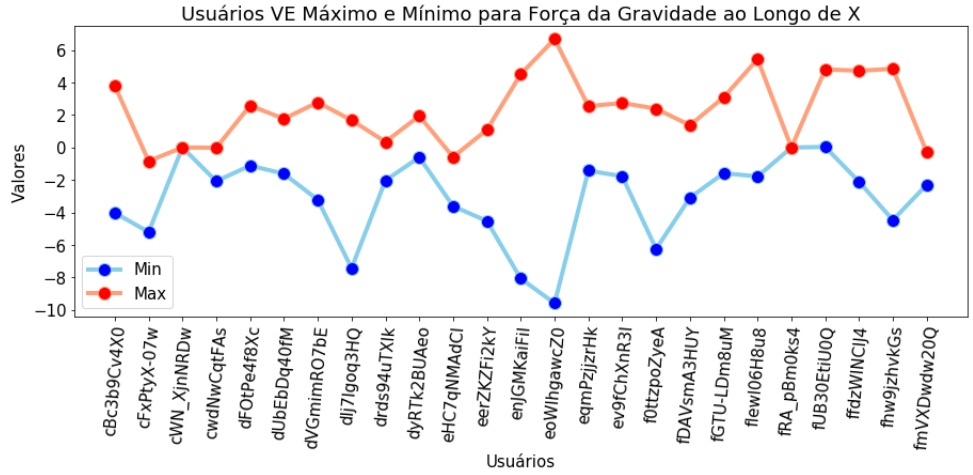

Figura 4.27: Gráfico VE valores máximos e mínimos para força da gravidade ao longo de X.

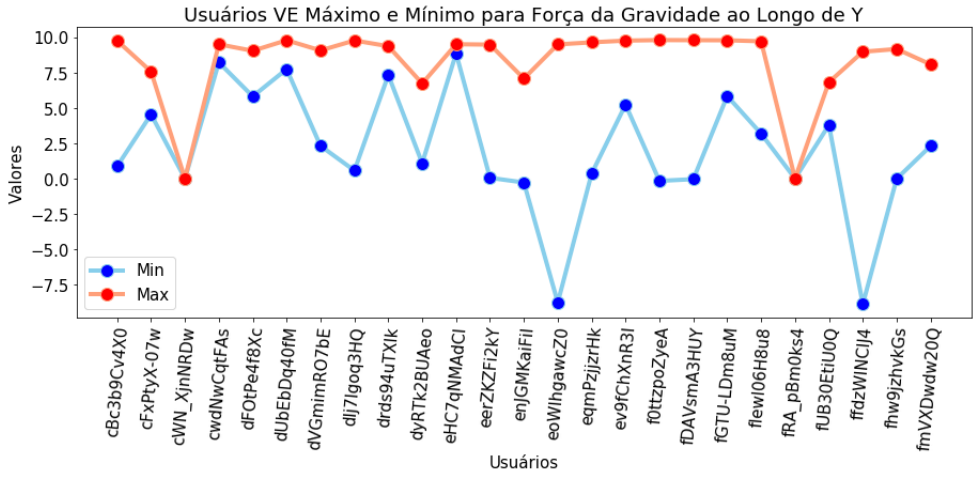

Figura 4.28: Gráfico VE valores máximos e mínimos para força da gravidade ao longo de Y.

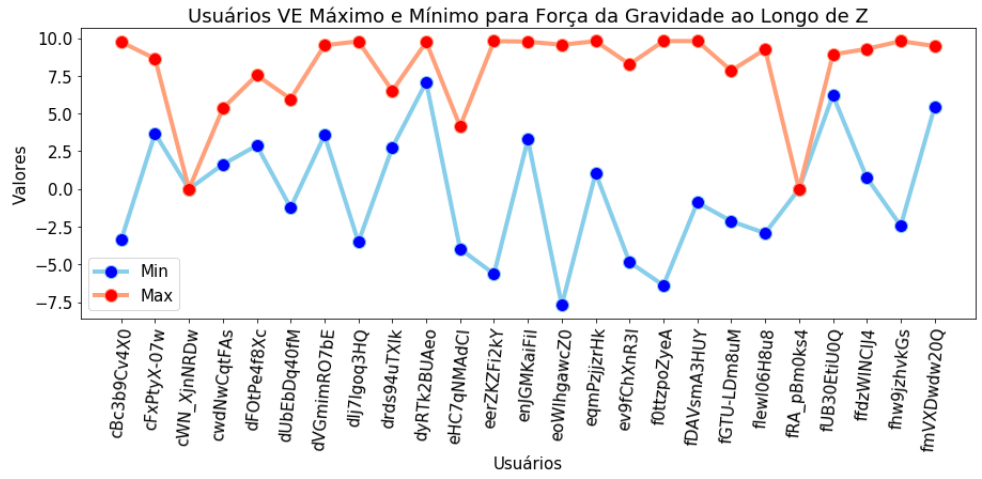

Figura 4.29: Gráfico VE valores máximos e mínimos para força da gravidade ao longo de Z.

4.4.2 Comparação Verificação Dinâmica

4.4.2.1 *Touchscreen*

Os valores de tempo pressiona pressiona em VD tiveram uma variação mais acentuada para três usuários, os outros apresentaram valores próximos, e em muitos casos aparentemente iguais, conforme demonstrado na Figura 4.30. Para dois usuários estão bem maiores em relação aos outros podendo indicar a presença de *outiliers* nos dados destes usuários.

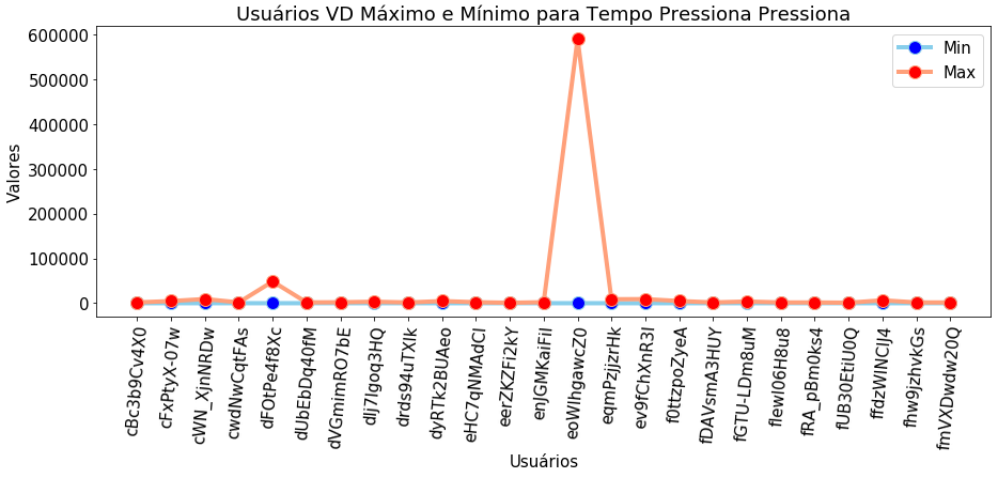

Figura 4.30: Gráfico VD valores máximos e mínimos para tempo pressiona pressiona.

Os valores de tempo pressiona solta em VD tiveram uma alta variação entre os usuários, conforme demonstrado na Figura 4.31.

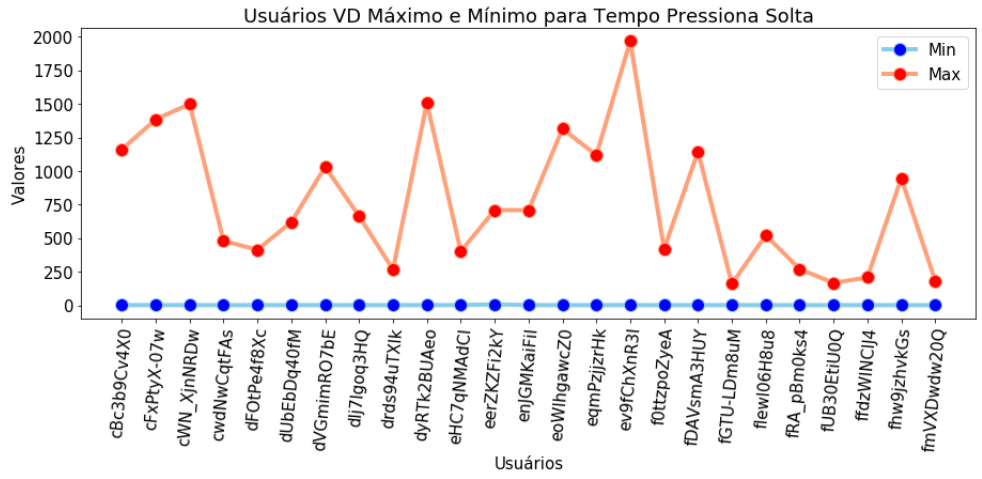

Figura 4.31: Gráfico VD valores máximos e mínimos para tempo pressiona solta.

Os valores de tempo solta pressiona em VD tiveram no geral uma baixa variação entre os usuários, e formou um gráfico bem semelhante ao do tempo pressiona pressiona, conforme demonstrado na Figura 4.32.

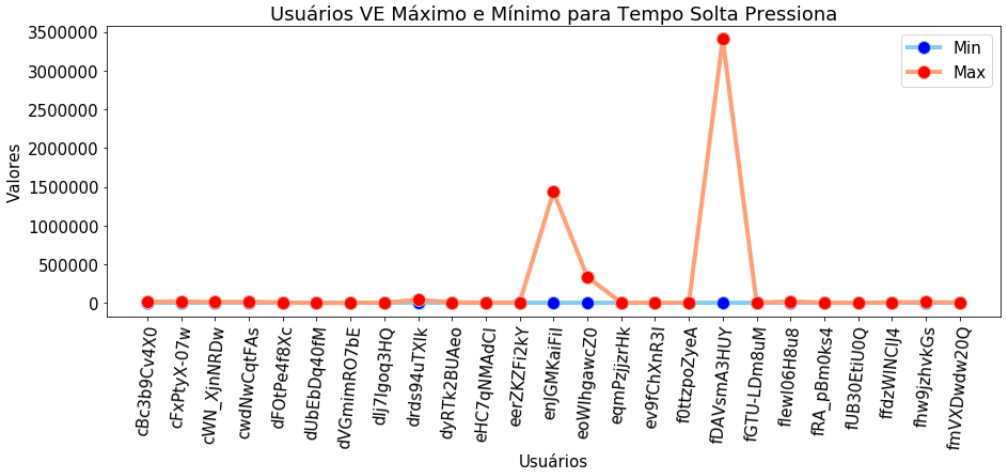

Figura 4.32: Gráfico VD valores máximos e mínimos para tempo solta pressiona.

Os valores de tempo solta solta em VD tiveram no geral uma baixa variação entre os usuários, e formou um gráfico bem semelhante ao do tempo pressiona pressiona e ao do solta pressiona, fazendo com que se chegue a conclusão que estas características podem não ser tão relevantes para a criação dos modelos, conforme demonstrado na Figura 4.33.

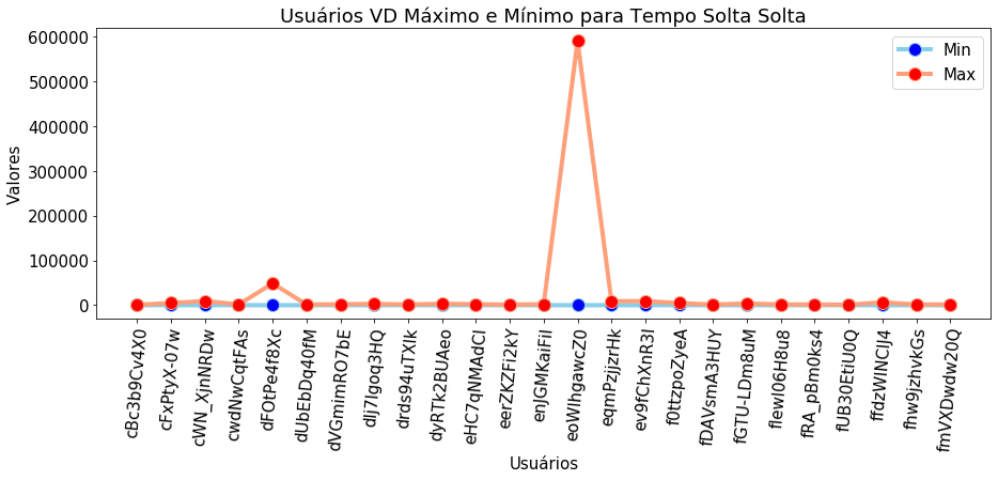

Figura 4.33: Gráfico VD valores máximos e mínimos para tempo solta solta.

Os valores de média do tempo de pressionamento em VD tiveram uma alta variação entre os usuários, assim como aconteceu para o pressiona solta, pois o tempo de pressionamento é uma característica derivada da do pressiona solta, conforme demonstrado na Figura 4.34.

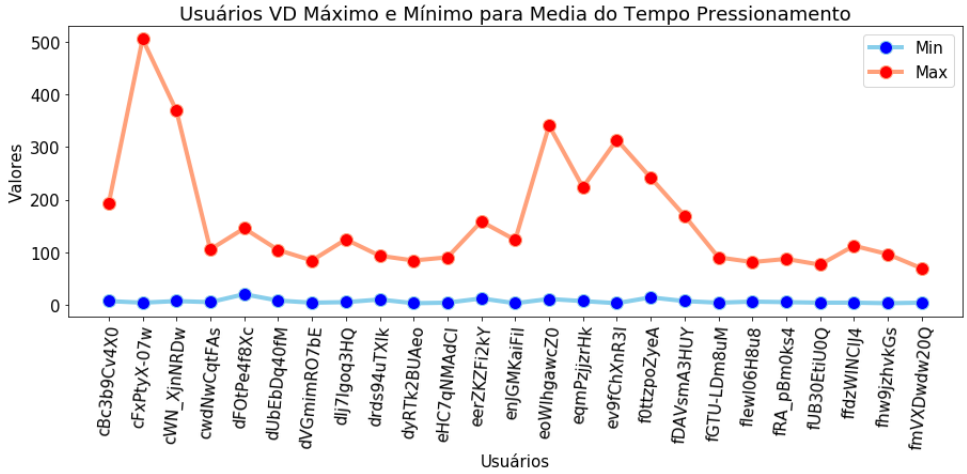

Figura 4.34: Gráfico VD valores máximos e mínimos para média do tempo de pressionamento.

Os valores de pressão para VD teve uma variação maior para 7 usuários, os outros usuários apresentaram valores máximos e mínimos próximos, conforme demonstrado na Figura 4.35.

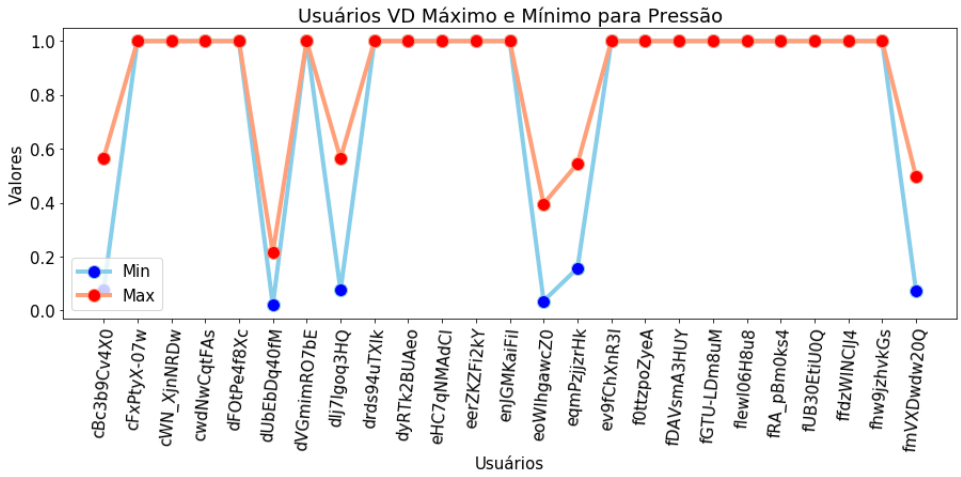

Figura 4.35: Gráfico VD valores máximos e mínimos para pressão.

Como os valores de pressão tiveram uma alta variação entre o máximo de mínimo dos usuários, fazendo com que não seja fácil visualizar os valores na Figura 4.35, com o objetivo de deixar mais clara a variação dos valores entre os usuários foi criada a Tabela 4.82, onde pode ser observado que para a grande maioria o valor retornado é 1, indicando que o sensor apenas consegue retornar o valor 1.

Tabela 4.82: Valores máximos e mínimos VD pressão.

Usuário	**Presão**	
Identificação	**Valor Mínimo**	**Valor Máximo**
drds94uTXlk	1,000000	1,000000
cwdNwCqtFAs	1,000000	1,000000
eHC7qNMAdCI	1,000000	1,000000
dUbEbDq40fM	0,019608	0,215686
fUB30EtiU0Q	1,000000	1,000000
cWN_XjnNRDw	1,000000	1,000000
ffdzWINClJ4	1,000000	1,000000
dFOtPe4f8Xc	1,000000	1,000000
cBc3b9Cv4X0	0,075000	0,565000
ev9fChXnR3I	1,000000	1,000000
fRA_pBm0ks4	1,000000	1,000000
dyRTk2BUAeo	1,000000	1,000000
cFxPtyX-07w	1,000000	1,000000
dVGmimRO7bE	1,000000	1,000000
fGTU-LDm8uM	1,000000	1,000000
fmVXDwdw20Q	0,074510	0,498039
eqmPzjjzrHk	0,156863	0,545098
enJGMKaiFiI	1,000000	1,000000

Identificação	Valor Mínimo	Valor Máximo
fIewI06H8u8	1,000000	1,000000
eerZKZFi2kY	1,000000	1,000000
dlj7Igoq3HQ	0,075000	0,565000
fDAVsmA3HUY	1,000000	1,000000
f0ttzpoZyeA	1,000000	1,000000
eoWIhgawcZ0	0,035294	0,396078
fhw9jzhvkGs	1,000000	1,000000

Os valores de média da pressão para VD apresentaram uma variação maior em relação a pressão, demonstrando que os usuários aplicam em média uma pressão diferente na interação com uma mesma aplicação *mobile*, mas para alguns usuários os valores ainda foram próximos, conforme demonstrado na Figura 4.36.

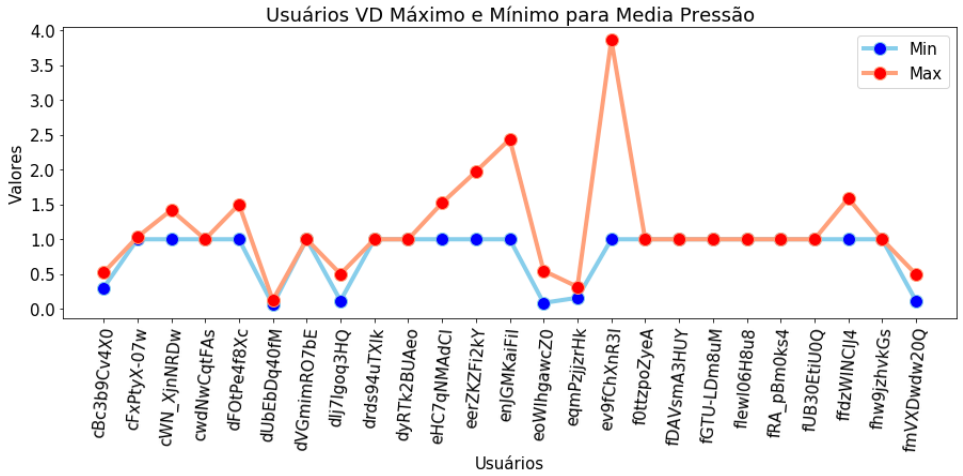

Figura 4.36: Gráfico VD valores máximos e mínimos para média pressão.

Os valores de tamanho do dedo para VD apresentaram uma variação entre os usuários, para dois usuários os valores de máximo ficaram bem acima dos demais, o que prejudicou a melhor visualização da variação dos valores de máximo e mínimo dos outros 23 usuários, conforme demonstrado na Figura 4.37.

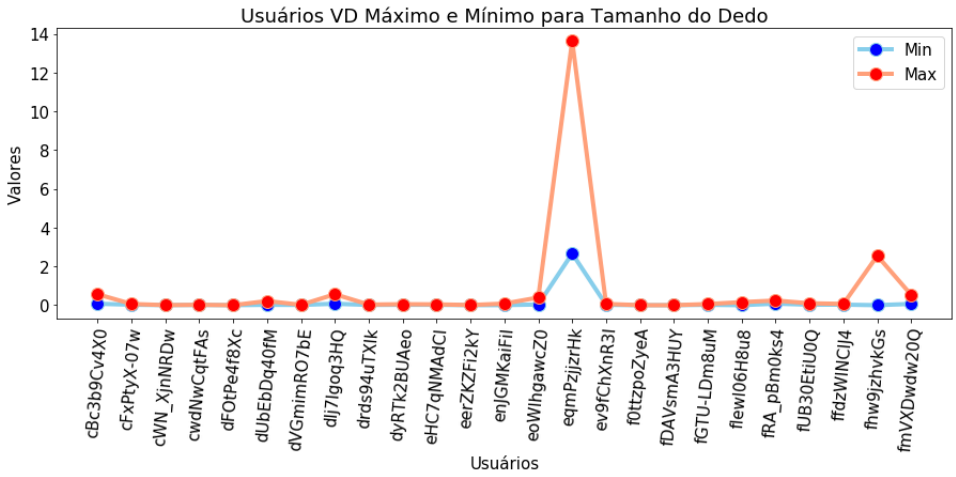

Figura 4.37: Gráfico VD valores máximos e mínimos para tamanho do dedo.

Como os valores do tamanho do dedo tiveram uma variação de valores máximos e mínimos muito grande entre os usuários, impossibilitando a correta visualização da diferença para esses valores na Figura 4.37, foi criada a Tabela 4.83 para detalhar a diferença de valores máximos e mínimos entre os usuários de forma mais clara, o que deixou evidente que os valores são realmente bem distintos entre os usuários.

Tabela 4.83: Valores máximos e mínimos VE tamanho do dedo.

Usuário	**Tamanho do dedo**	
Identificação	**Valor Mínimo**	**Valor Máximo**
drds94uTXlk	0,011765	0,031373
cwdNwCqtFAs	0,011765	0,019608
eHC7qNMAdCI	0,019608	0,043137
dUbEbDq40fM	0,019608	0,215686
fUB30EtiU0Q	0,023529	0,098039
cWN_XjnNRDw	0,003922	0,003922
ffdzWINClJ4	0,027451	0,066667
dFOtPe4f8Xc	0,000000	0,000000
cBc3b9Cv4X0	0,075000	0,565000
ev9fChXnR3I	0,019608	0,066667
fRA_pBm0ks4	0,080000	0,240000
dyRTk2BUAeo	0,019608	0,050980
cFxPtyX-07w	0,023529	0,062745
dVGmimRO7bE	0,011765	0,019608
fGTU-LDm8uM	0,019608	0,058824
fmVXDwdw20Q	0,066667	0.533333
eqmPzjjzrHk	2,666667	13,633334
enJGMKaiFiI	0,003922	0,082353

Identificação	Valor Mínimo	Valor Máximo
fIewI06H8u8	0,000000	0,158824
eerZKZFi2kY	0,000000	0,007874
dlj7Igoq3HQ	0,075000	0,565000
fDAVsmA3HUY	0,000000	0,000000
f0ttzpoZyeA	0,000000	0,000000
eoWIhgawcZ0	0,035294	0,396078
fhw9jzhvkGs	0,000000	2,550000

Os valores de média do tamanho do dedo para VD apresentaram uma variação entre os usuários, assim como aconteceu com a de tamanho do dedo dois usuários tiveram valores de máximo muito acima dos demais, o que prejudicou novamente a melhor visualização da variação dos valores de máximo e mínimo dos outros 23 usuários. Indicando que estes usuários tem um tamanho de área de contato do dedo com a tela bem maior que os outros, conforme demonstrado na Figura 4.38.

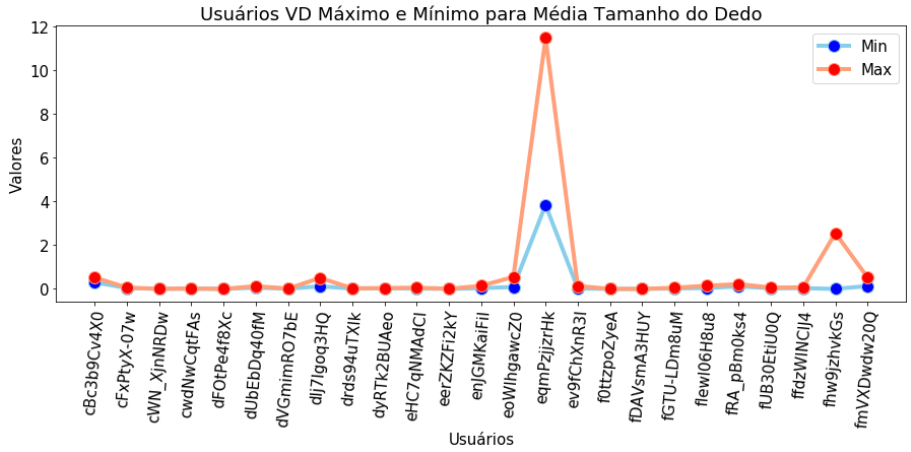

Figura 4.38: Gráfico VD valores máximos e mínimos para média do tamanho do dedo.

Os valores para as coordenadas X e Y em VD tiveram uma alta variação, conforme demonstrado nas Figuras 4.39 e 4.40.

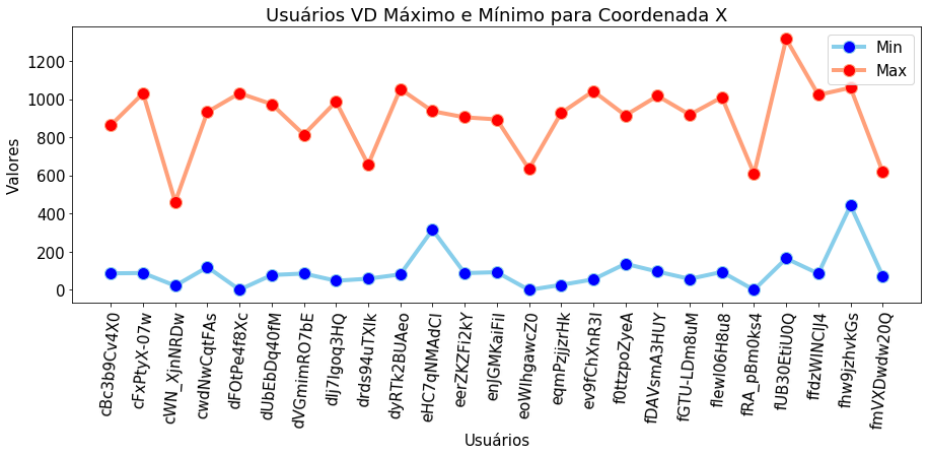

Figura 4.39: Gráfico VD valores máximos e mínimos para coordenada X.

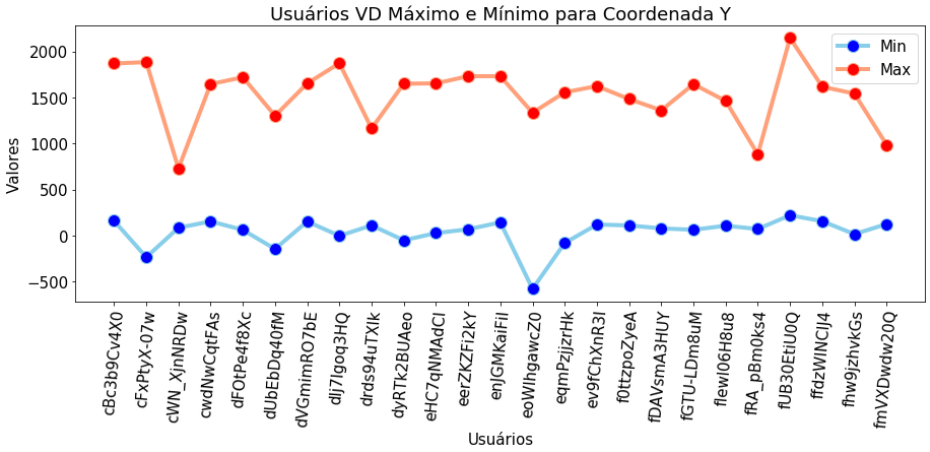

Figura 4.40: Gráfico VD valores máximos e mínimos para coordenada Y.

4.4.2.2 Acelerômetro

Os valores de aceleração ao longo do eixo X, Y e Z em VD, geradas pelo acelerômetro, tiveram uma alta variação entre os usuários, indicando que estas podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.41, 4.42 e 4.43.

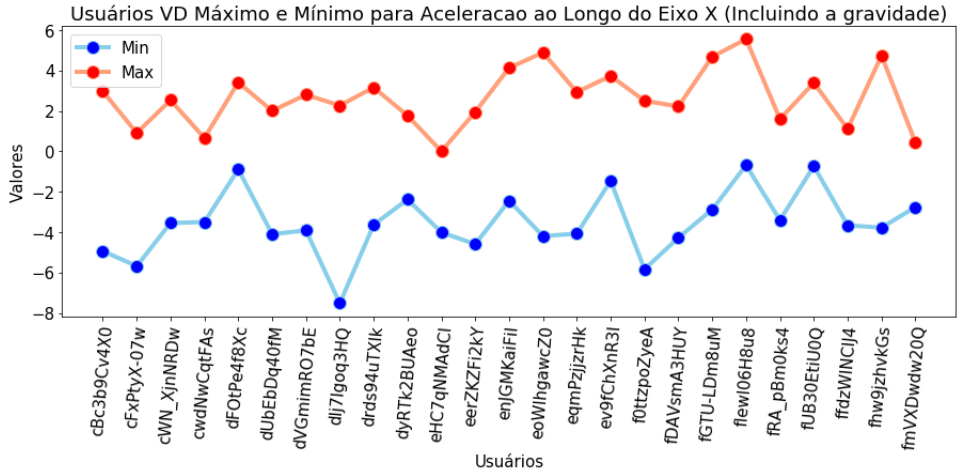

Figura 4.41: Gráfico VD valores máximos e mínimos para aceleração ao longo do eixo X (incluindo a gravidade).

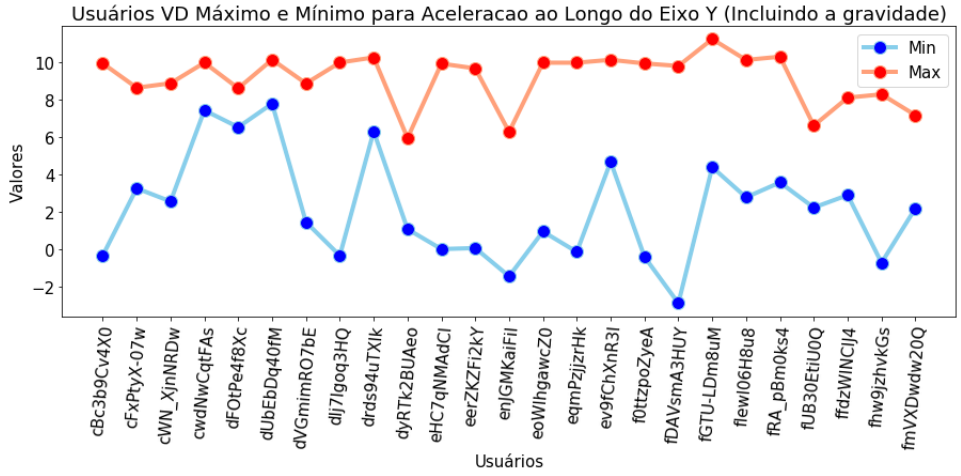

Figura 4.42: Gráfico VD valores máximos e mínimos para aceleração ao longo do eixo Y (incluindo a gravidade).

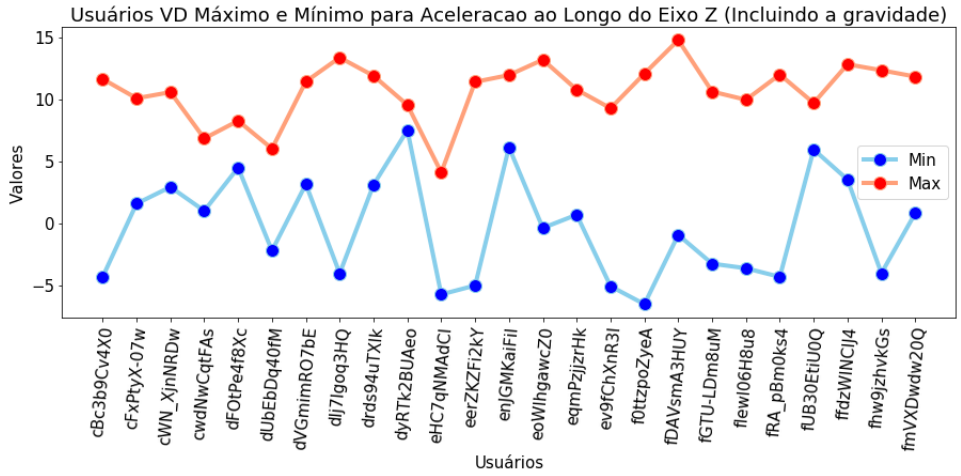

Figura 4.43: Gráfico VD valores máximos e mínimos para aceleração ao longo do eixo Z (incluindo a gravidade).

4.4.2.3 Giroscópio

Os valores de rotação ao redor de X, Y e Z em VD, geradas pelo giroscópio, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.44, 4.45, e 4.46.

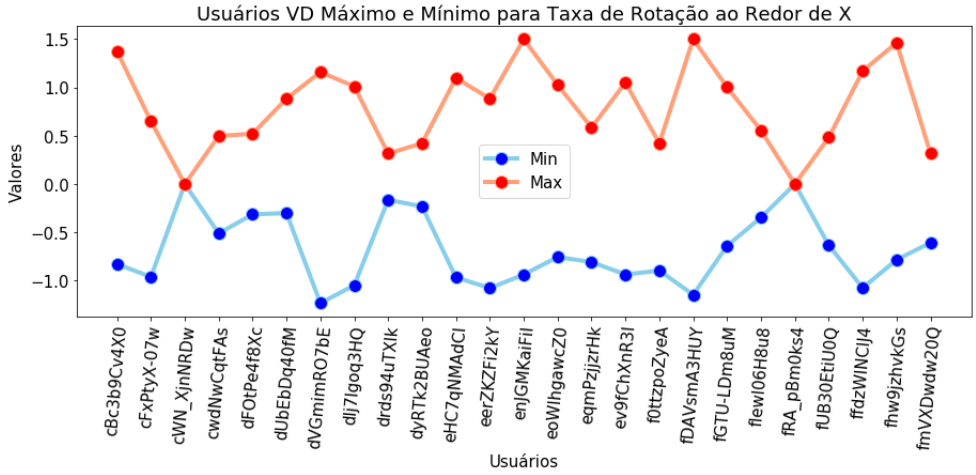

Figura 4.44: Gráfico VD valores máximos e mínimos para taxa de rotação ao redor de X.

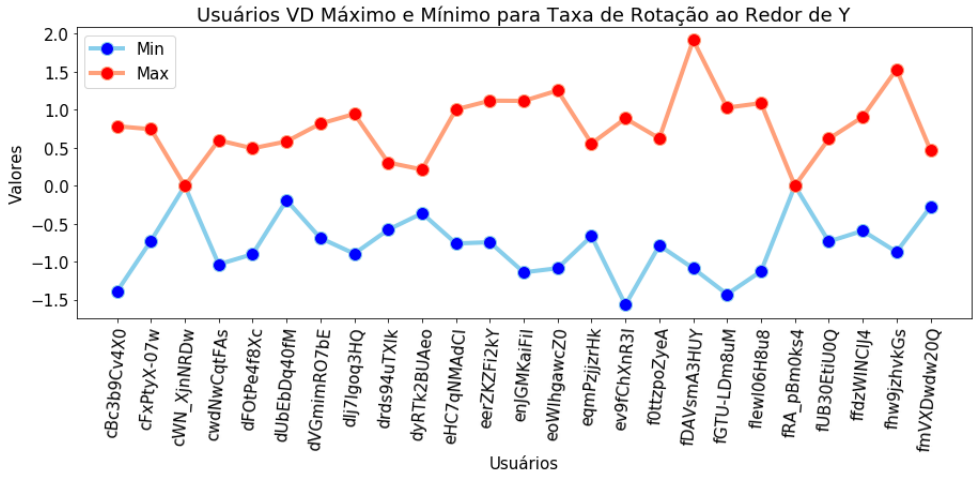

Figura 4.45: Gráfico VD valores máximos e mínimos para taxa de rotação ao redor de Y.

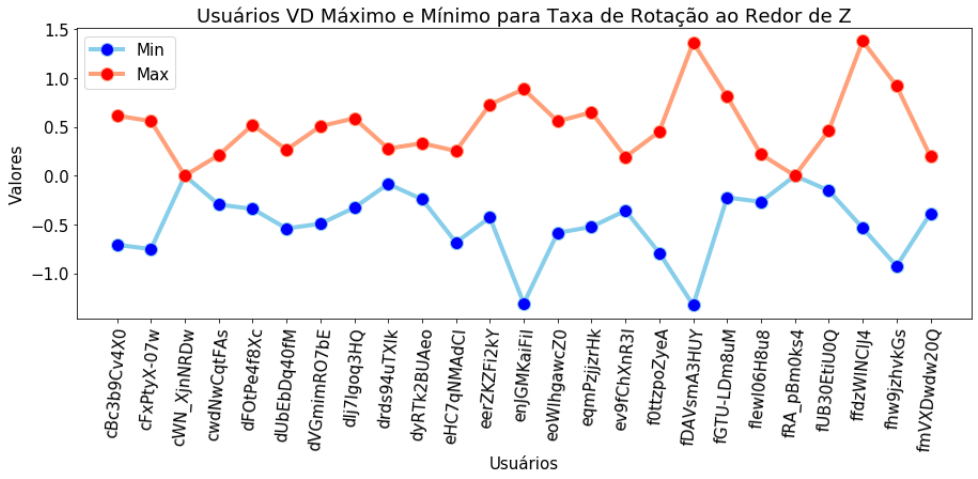

Figura 4.46: Gráfico VD valores máximos e mínimos para taxa de rotação ao redor de Z.

4.4.2.4 Magnetômetro

Os valores para o campo geomagnético sobre X, Y e Z em VD, geradas pelo magnetômetro, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.47, 4.48, e 4.49.

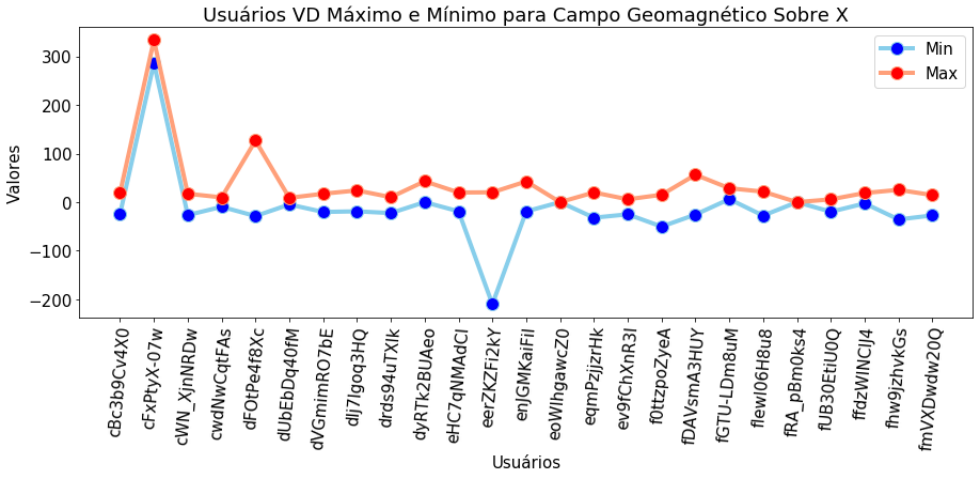

Figura 4.47: Gráfico VD valores máximos e mínimos para campo geomagnético sobre X.

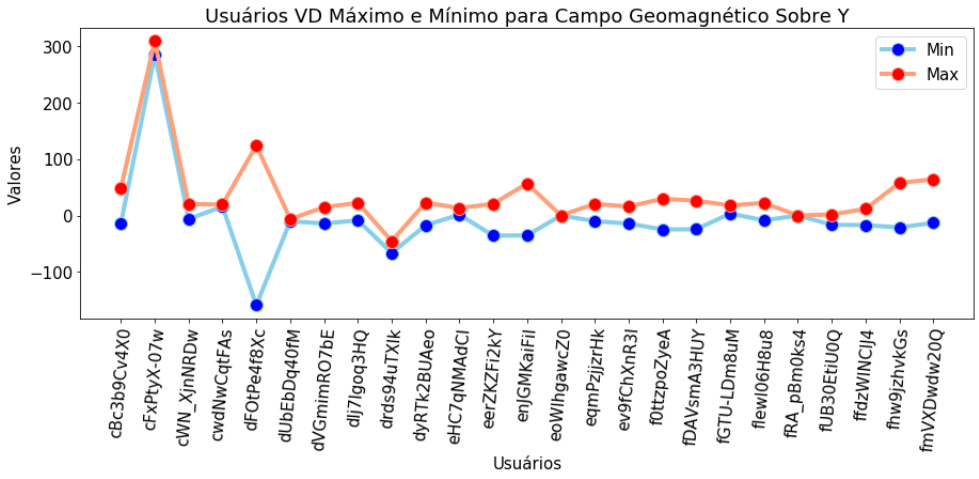

Figura 4.48: Gráfico VD valores máximos e mínimos para campo geomagnético sobre Y.

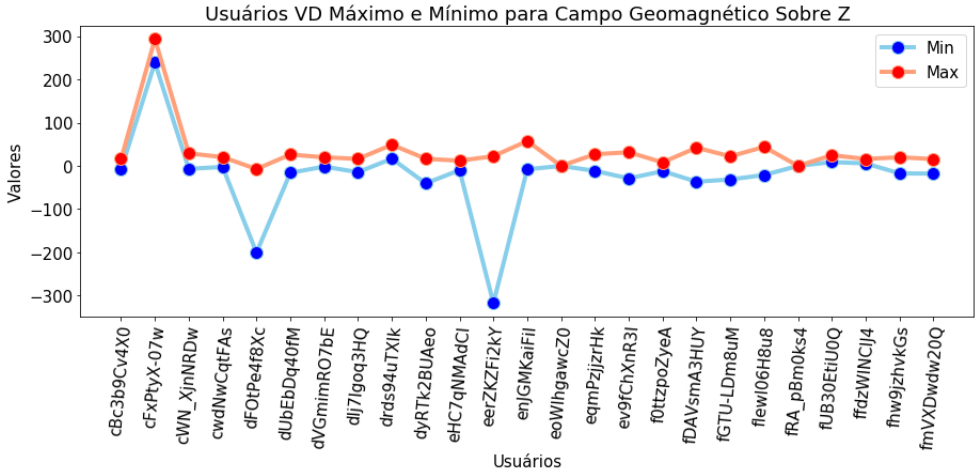

Figura 4.49: Gráfico VD valores máximos e mínimos para campo geomagnético sobre Z.

4.4.2.5 Sensores de Rotação

No caso das características geradas a partir dos sensores de rotação em VD, a acurácia estimada foi o que apresentou um grau menor de variação para máximos e mínimos entre os usuários. Já os valores para o componente escalar vetor rotação, e os componente do vetor de rotação ao longo de X, Y e Z, tiveram uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.50, 4.51, 4.52, 4.53, e 4.54.

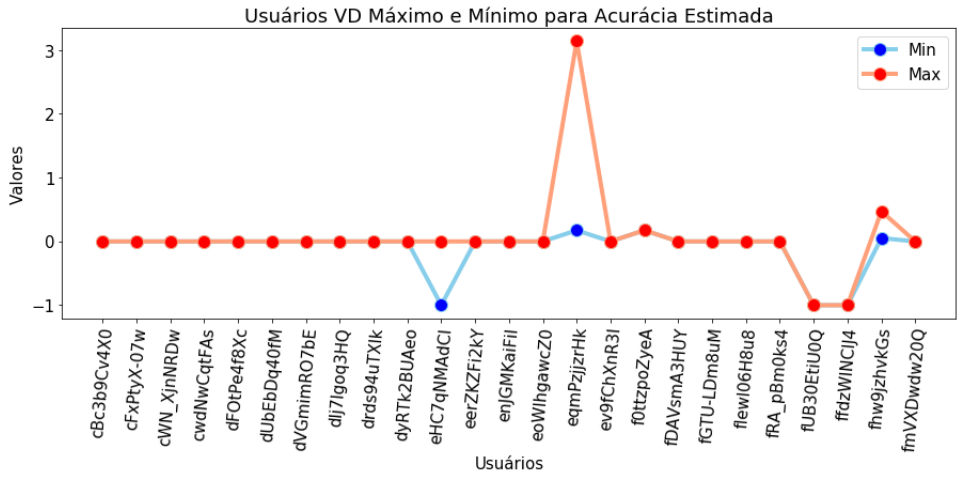

Figura 4.50: Gráfico VD valores máximos e mínimos para acurácia estimada.

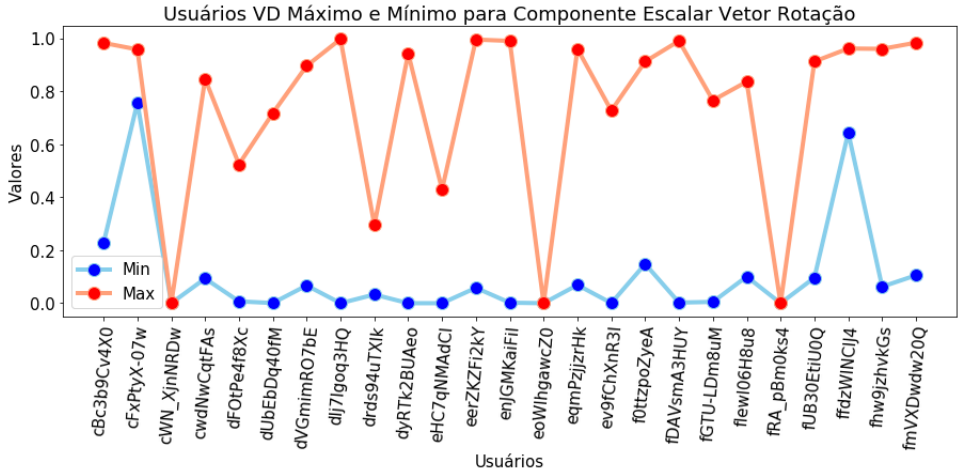

Figura 4.51: Gráfico VD valores máximos e mínimos para componente escalar vetor rotação.

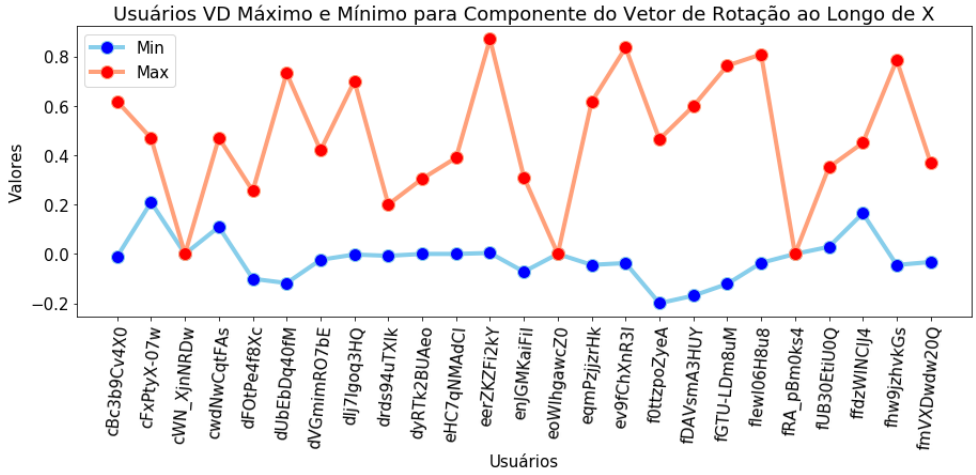

Figura 4.52: Gráfico VD valores máximos e mínimos para componente do vetor de rotação ao longo de X.

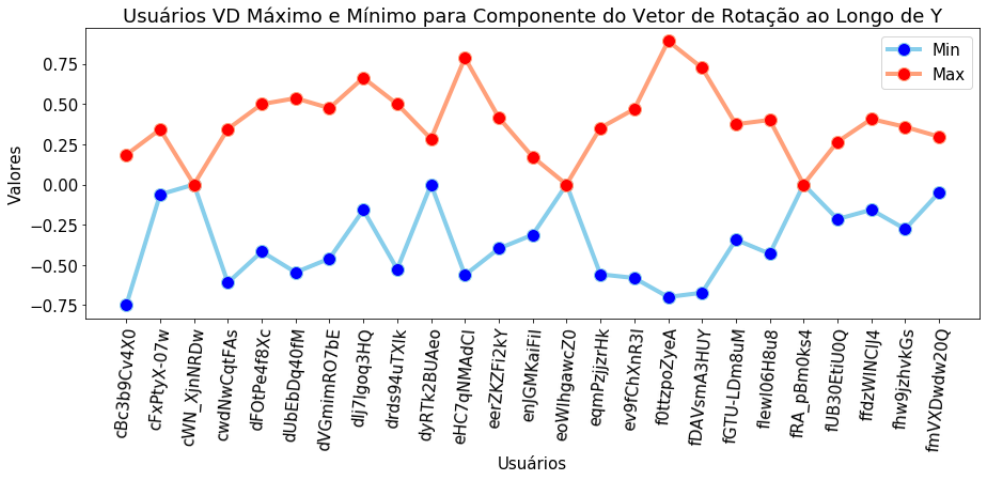

Figura 4.53: Gráfico VD valores máximos e mínimos para componente do vetor de rotação ao longo de Y.

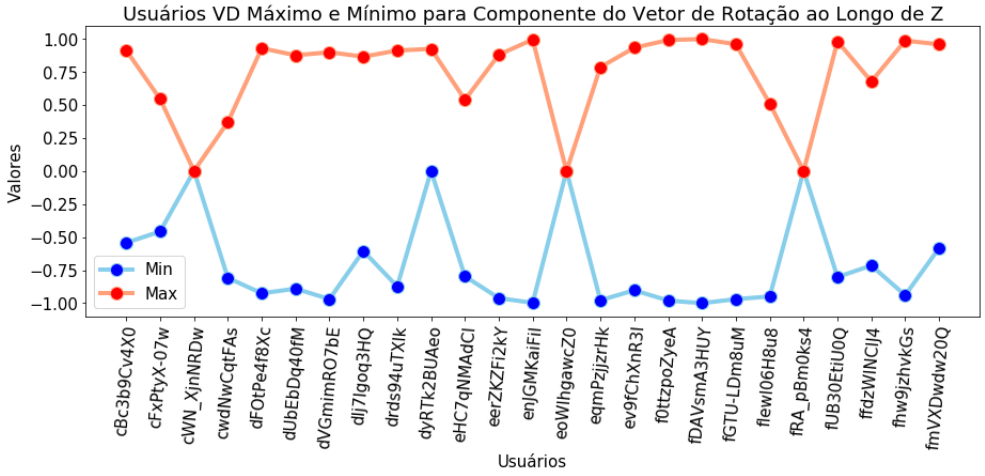

Figura 4.54: Gráfico VD valores máximos e mínimos para componente do vetor de rotação ao longo de Z.

4.4.2.6 Sensores de Aceleração

Os valores para força da aceleração longo de X, Y e Z em VD, excluindo a gravidade, geradas pelos sensores de aceleração, variou também entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.55, 4.56, e 4.57.

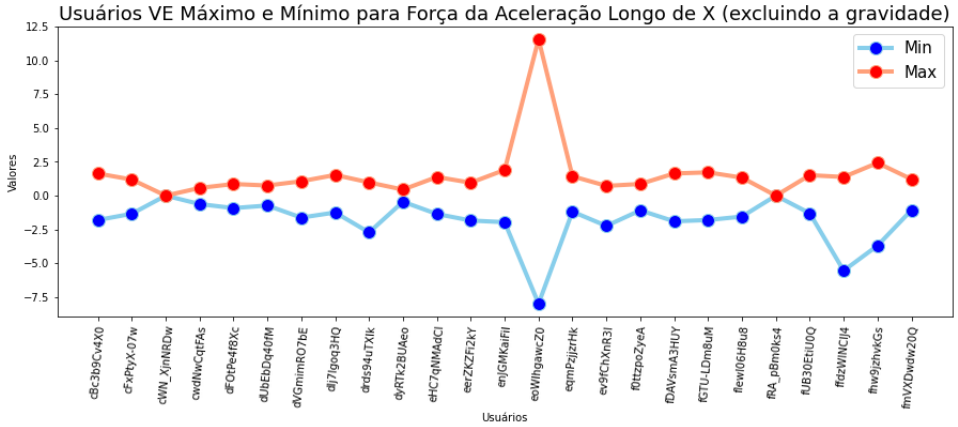

Figura 4.55: Gráfico VD valores máximos e mínimos para força da aceleração longo de X (excluindo a gravidade).

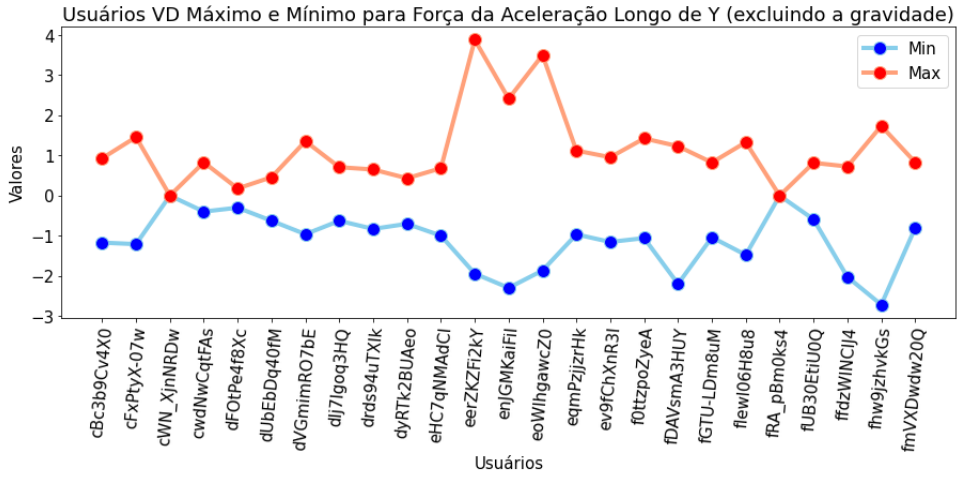

Figura 4.56: Gráfico VD valores máximos e mínimos para força da aceleração longo de Y (excluindo a gravidade).

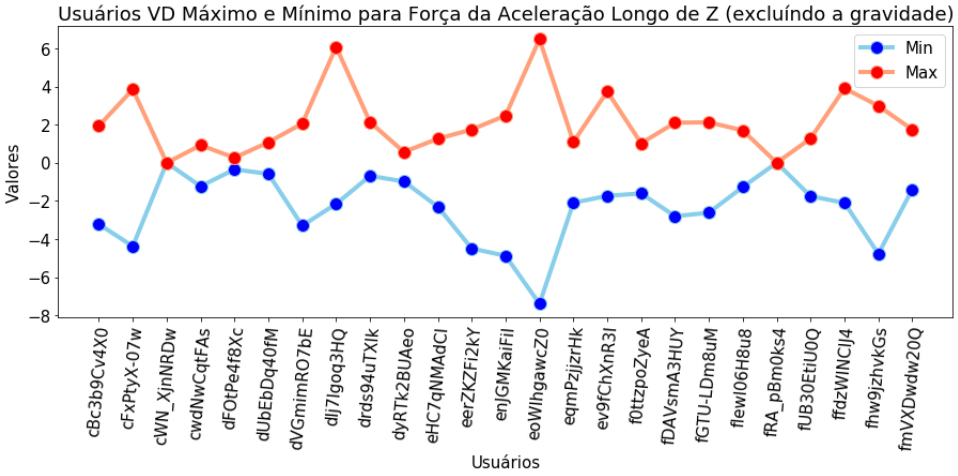

Figura 4.57: Gráfico VD valores máximos e mínimos para força da aceleração longo de Z (excluindo a gravidade).

4.4.2.7 Sensores de Gravidade

Os valores para força da gravidade ao longo de X, Y e Z em VD, geradas pelos sensores de gravidade, tiveram também uma alta variação entre os usuários, indicando que estas também podem ser características de forte discriminação entre os usuários, conforme demonstrado nas Figuras 4.58, 4.59, e 4.60.

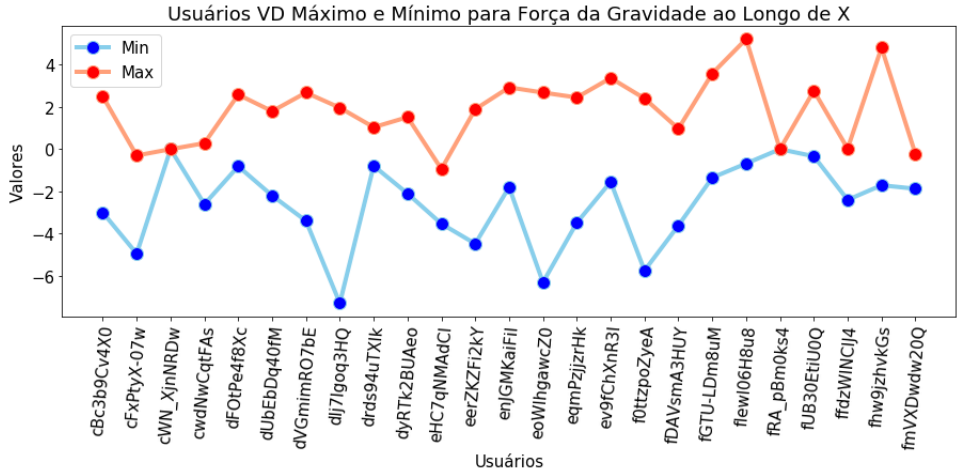

Figura 4.58: Gráfico VD valores máximos e mínimos para força da gravidade ao longo de X.

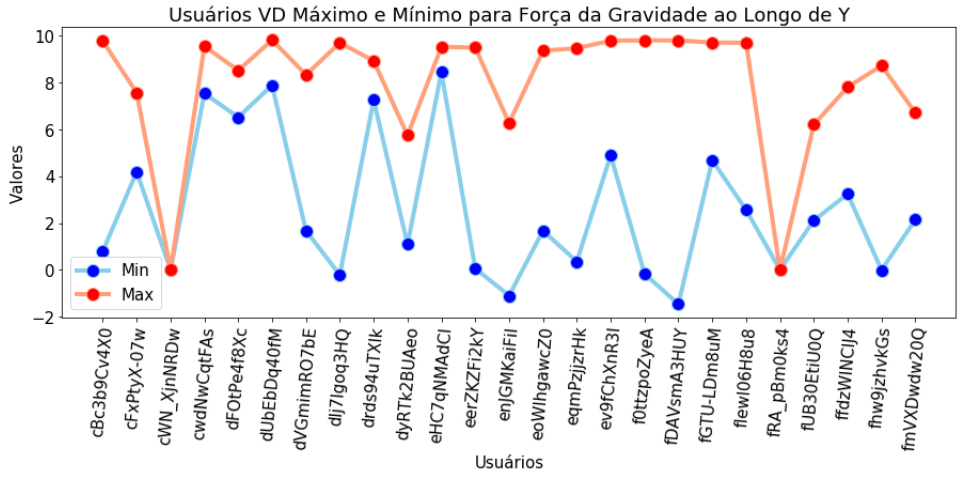

Figura 4.59: Gráfico VD valores máximos e mínimos para força da gravidade ao longo de Y.

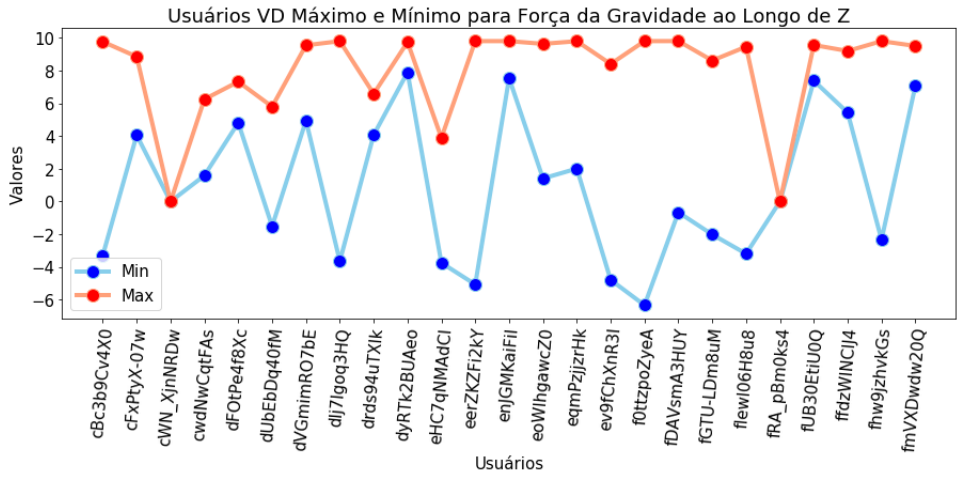

Figura 4.60: Gráfico VD valores máximos e mínimos para força da gravidade ao longo de Z.

4.5 CARACTERÍSTICAS MAIS IMPORTANTES

A fim de verificar quais as características mais importantes para a geração dos modelos utilizou-se o algoritmo RF, tanto para a verificação estática quanto dinâmica para gerar os rankings das características mais importantes conforme Tabelas 4.84 e 4.85.

Tabela 4.84: Ranking das características por importância para o Momento 1.

Característica	**Valor de importância Momento 1**
Tamanho do dedo	0,240485
Média do Tamanho do Dedo	0,236985
Pressão	0,065277
Média da Pressão	0,060130
Campo Geomagnético Sobre Y	0,058717
Campo Geomagnético Sobre X	0,053064
Campo Geomagnético Sobre Z	0,044218
Media do Tempo Pressionamento	0,042234
Componente Escalar Vetor Rotação	0,036942
Aceleração ao Longo do Eixo Y	0,031548
Força da gravidade ao longo de Y	0,029639
Forca da Gravidade ao Longo de Z	0,022126
Acurácia Estimada	0,018292
Força da gravidade ao longo de X	0,017382
Componente do vetor de rotação ao longo de Z	0,016680
Componente do vetor de rotação ao longo de X	0,012581
Componente do vetor de rotação ao longo de Y	0,009849

Característica	Valor de importância Momento 1
Aceleração ao Longo do Eixo X	0,003132
Aceleração ao Longo do Eixo Z	0,000566
Tempo pressiona pressiona	0,000047
Tempo solta pressiona	0,000039
Taxa de rotação ao redor de Z	0,000035
Força da aceleração longo de Z (excluindo a gravidade)	0,000027
Força da aceleração longo de X (excluindo a gravidade)	0,000003
Tempo pressiona solta	0,000002
Taxa de rotação ao redor de X	0,000000
Taxa de rotação ao redor de y	0,000000
Tempo solta solta	0,000000
Força da aceleração longo de Y (excluindo a gravidade)	0,000000

Conforme verificado da Tabela 4.84, os dados de sensores que foram mais importantes para o Momento 1 foram: magnetômetro, sensor de rotação, acelerômetro e sensor da gravidade. O magnetômetro teve todas as suas três informações entre as 10 mais importantes, demonstrando que a forma como o usuário dispõe o dispositivo no espaço físico é uma informação com forte discriminância entre os usuários.

Tabela 4.85: Ranking das características por importância para o Momento 2.

Característica	Valor de importância Momento 2
Média do Tamanho do Dedo	0,268114
Tamanho do dedo	0,138374
Componente Escalar Vetor Rotação	0,090839
Pressão	0,090122
Média da Pressão	0,087559
Campo Geomagnético Sobre Y	0,056308
Campo Geomagnético Sobre X	0,051553
Componente do vetor de rotação ao longo de X	0,034605
Força da gravidade ao longo de Y	0,030806
Campo Geomagnético Sobre Z	0,025307
Aceleração ao Longo do Eixo Y	0,023385
Forca da Gravidade ao Longo de Z	0,022277
Acurácia Estimada	0,019247

Característica	**Valor de importância Momento 2**
Media do Tempo Pressionamento	0,012550
coordenadaY	0,009005
Força da gravidade ao longo de X	0,006908
Componente do vetor de rotação ao longo de Y	0,006821
Aceleração ao Longo do Eixo Z	0,006500
Componente do vetor de rotação ao longo de Z	0,006133
Tempo solta solta	0,005452
coordenadaX	0,005168
Aceleração ao Longo do Eixo X	0,002446
Força da aceleração longo de X (excluindo a gravidade)	0,000337
Tempo pressiona pressiona	0,000093
Força da aceleração longo de Z (excluíindo a gravidade)	0,000084
Tempo solta pressiona	0,000005
Taxa de rotação ao redor de X	0,000000
Taxa de rotação ao redor de y	0,000000
Taxa de rotação ao redor de Z	0,000000
Tempo pressiona solta	0,000000
Força da aceleração longo de Y (excluindo a gravidade)	0,000000

Para o Momento 2, conforme Tabela 4.85, os dados de sensores mais importantes foram: sensor de rotação, magnetômetro, sensor de rotação e sensor de gravidade. Novamente o magnetômetro teve todas as suas três informações entre as 10 mais importantes.

Nas Figuras 4.61 e 4.62, são listadas as 10 características mais importantes para os Momentos 1 e 2.

A Figura 4.61 detalha as 10 características mais importantes para o Momento 1 e Figura 4.62 detalha as 10 características mais importantes para o Momento 2.

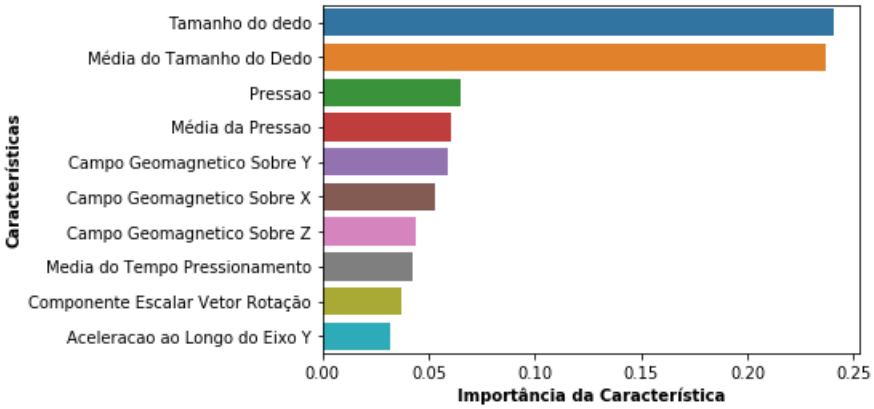

Figura 4.61: Ranking das 10 características mais importantes VE.

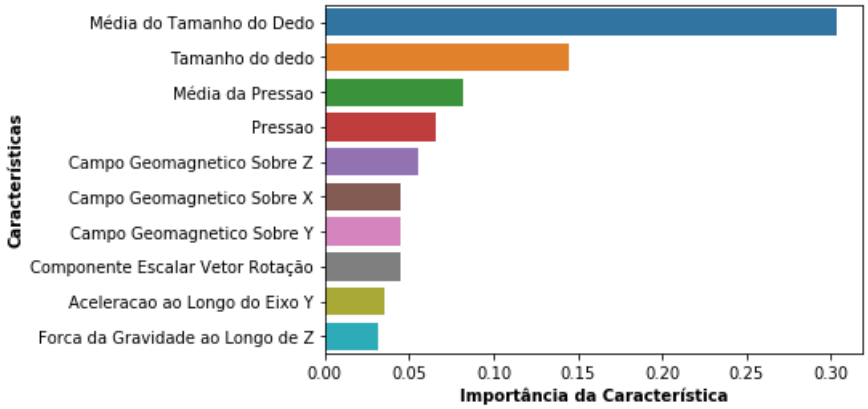

Figura 4.62: Ranking das 10 características mais importantes VD.

Conforme demonstram as Tabelas 4.84 e 4.85 e as Figuras 4.61 e 4.62, as características mais discriminantes para ambos os Momentos são: o tamanho do dedo e a média do tamanho do dedo, sendo assim a qualidade destes dados é muito importante para a criação dos modelos. Essa foi umas das justificativas para a criação do Escopo C, para que fosse possível analisar o desempenho dos modelos quando excluídas essas características. Os dados de sensores contribuiram para a definição do padrão dos usuários, pois tanto para as verificações estática e dinâmica, de 40 a 60% das 10 características mais importantes foram coletadas dos sensores abrangidos no experimento.

4.6 LOCALIZAÇÃO

Para a definição do padrão de localização, que no *framework* proposto compõe o resultado final de fusão do *score* para os Momentos 1 e 2, o algoritmo SVM *One Class* foi treinado com os valores de latitude e longitude coletados de cada usuário durante as sessões de interação com a aplicação. Para a criação do modelo foram utilizados todos os dados de latitude e longitude coletados durante os Momentos 1 e 2, pois é tido o pressuposto que o algoritmo já tenha o conhecimento de todos os valores de latitude e longitude do usuário. Para criação do modelo foram

utilizados apenas os valores únicos, visando reduzir o tamanho dos dados de treino e tornando mais simples e rápida a fase treino. Para teste foram utilizados os valores de latitude e longitude presentes nos *templates*, de acordo com a quantidade definida para os cenários de 1 a 3, para VE e VD. Os resultados de acurácia por usuário podem ser conferidos na Tabela 4.86.

Tabela 4.86: Acurácia por usuário para o modelo de localização entre os cenários

Usuário	**Localização acurácia por Cenário**		
Identificação	**Acurácia C1**	**Acurácia C2**	**Acurácia C3**
drds94uTXlk	92,30%	—	—
cwdNwCqtFAs	90,62%	—	—
eHC7qNMAdCI	91,54%	—	—
dUbEbDq40fM	92,3%	—	—
fUB30EtiU0Q	92,91%	—	—
cWN_XjnNRDw	93,10%	—	—
ffdzWINClJ4	93,00%	—	—
dFOtPe4f8Xc	91,89	88,0%	—
cBc3b9Cv4X0	89,47%	93,33%	—
ev9fChXnR3I	93,29%	92,85%	—
fRA_pBm0ks4	92,82%	92,82%	—
dyRTk2BUAeo	91,42%	90,56%	91,66%
cFxPtyX-07w	95,83%	95,00%	—
dVGmimRO7bE	94,11%	91,42%	—
fGTU-LDm8uM	92,79%	92,85%	92,76%
fmVXDwdw20Q	92,61%	92,76%	92,30%
eqmPzjjzrHk	93,41%	93,19%	91,07%
enJGMKaiFiI	90,80%	89,85%	89,47%
fIewI06H8u8	92,60%	92,15%	92,50%
eerZKZFi2kY	90,00%	88,70%	93,75%
dlj7Igoq3HQ	92,18%	89,58%	89,47%
fDAVsmA3HUY	92,62%	92,70%	92,56%
f0ttzpoZyeA	92,55%	92,63%	91,73%
eoWIhgawcZ0	92,95%	91,93%	91,39%
fhw9jzhvkGs	93,18%	92,57%	92,13%

O algoritmo SVM *One Class* apresentou acurácia a partir de 90%, ou bem próxima deste valor, sempre que treinado com todas as informações de latitude e longitude do usuário, demonstrando assim a eficiência do algoritmo para a definição do padrão de localização de um indivíduo para o limiar definido. E foi notado que a apresentação de acurácia menor que o limiar é um indicativo de utilização do aplicativo em um novo local, gerando a necessidade de retreinar o modelo para o

usuário.

4.7 FUSÃO DOS RESULTADOS

O último passo para a geração do *score* final no *framework* proposto é unir os valores gerados de acurácia para a localização, com os valores de F1 gerados para os Momentos 1 e 2. Para representar o resultado de fusão de *scores*, de acordo com a proposta do modelo, foi selecionado, para os usuários que tiveram um modelo criado para os Momentos 1 e 2 de acordo com o requisitos do *framework*, o primeiro resultado entre os cenários para VE e VD, e o valor da acurácia de localização levada em consideração será sempre a do cenário de maior número de acordo com o resultado de VE e VD. Essa é apenas uma ilustração, pois os resultados de VE e VD foram obtidos com a autenticação de vários *templates* de um mesmo usuário, mas em um cenário real a fusão dos *scores* será de resultados relacionados a apenas um *template* em tempo de execução para VE, VD e localização. Os resultados para essa ilustração são demonstrados na Tabela 4.87.

Tabela 4.87: Fusão dos resultados VE, VD e localização.

Usuário	**Fusão dos *Scores* por Usuário**					
Identificação	**Cenário VE**	**F1 VE**	**Cenário VD**	**F1 VD**	**Acurácia Localização**	**Resultado Fusão**
cwdNwCqtFAs	1	99,32	1	96,6	90,62	95,51
dUbEbDq40fM	1	94,33	1	94,25	92,3	93,62
fRA_pBm0ks4r	1	100	2	100	92,82	97,6
cFxPtyX-07w	1	100	2	100	92,82	97,6
dVGmimRO7bE	2	99,78	1	96,54	95,83	97,21
fGTU-LDm8uM	1	91,34	3	99,37	92,76	94,49
fmVXDwdw20Q	1	95,47	1	95,37	92,61	94,48
eqmPzjjzrHk	1	100	2	100	93,19	97,73
enJGMKaiFiI	1	97,85	3	90,2	89,47	92,5
fIewI06H8u8	1	99,11	3	92,07	92,5	94,56
dlj7Igoq3HQ	1	99,91	1	98,29	92,18	96,79
fDAVsmA3HUY	1	96,87	1	99,94	92,62	96,47
f0ttzpoZyeA	1	97,03	2	90,23	92,63	93,29
eoWIhgawcZ0	1	99,16	1	98,72	92,95	96,94

De acordo com os resultados demonstrados nesta seção, o *framework* proposto foi capaz de encontrar um modelo satisfatório para VE e VD, para 14 dos 25 usuários que participaram do experimento. Para VE com a maioria no cenário 1, já em VD houve uma distribuição maior entre os cenários, indicando que em VD são necessários, no geral, mais dados que em VE para criar um modelo de qualidade.

5 CONCLUSÃO E TRABALHOS FUTUROS

Os resultados dos experimentos desenvolvidos reforçam a tese de como é complexa a tarefa de encontrar um algoritmo de Aprendizado de Máquina que se possa generalizar para vários usuários, pois o padrão de interação com uma aplicação é único de cada indivíduo. Tornando importante a característica do *framework* proposto em utilizar uma gama de 6 algoritmos diferentes. Em relação aos objetivos propostos, o objetivo geral foi alcançado, pois foi implementado e validado com experimentos um *framework* capaz de executar a autenticação contínua de usuários baseado no seu comportamento biométrico na interação *touch* com uma aplicação *mobile* e seu padrão de localização.

Quanto aos objetivos específicos, de acordo com os cenários abordados no escopo proposto, unindo os escopos de EA, EB e ED, o resultado do experimento demonstrou que no cenário 1, com 5 interações com a aplicação, já é possível encontrar um algoritmo com F1 a partir de 90% para 88% dos usuários para verificação estática e para 60,86% no caso da verificação dinâmica. Isso indica que o valor mínimo de 5 *templates*, para VE e 15 para VD, pode ser considerado satisfatório como valor mínimo para treino, considerando as características da aplicação utilizada no experimento proposto neste trabalho. Foi observado também que os padrões se mantêm consistentes para os usuários entre as sessões, pois foi possível encontrar um modelo com o F1 desejado e FAR para impostores menor que 10% entre os cenários, para 80% dos usuários em VE e para 69,56% em VD. Foi observado também que em VE e VD, em relação aos usuários para os quais não foi possível encontrar um modelo que atendesse todos os requisitos definidos no *framework*, a maioria não ofereceu a quantidade de *templates* necessárias para o cenário 3.

Diante dos resultados observados nos escopos de A a F foi possível perceber que os dados coletados dos sensores são realmente importantes para a definição do padrão de autenticação biometrica *touch* de um usuário. Por exemplo, os escopos EB e EE demonstraram, no geral, uma performance pior em relação aos escopos que utilizam os dados de sensores. Outra constatação que reforçou a importância dos dados de sensores foi que esses estavam presentes entre as 10 características mais importantes com representação entre 40 e 60%, com destaque para as coletadas via magnetômetro que estavam, todas três, entre as 10 mais importantes tanto para VE quanto para VD.

Para o F1 e EER médios os resultados apresentados entre 90,68 e 97,05% e 9,85 e 1,88% respectivamente, entre os cenários do escopo proposto, validam a perspectiva promissora da utilização da biometria comportamental *touch* como uma aliada, se considerada em conjunto com métodos tradicionais como a senha, na definição de camadas de segurança para mitigação de fraudes ligadas à autenticação em aplicações bancárias *mobile*.

5.1 TRABALHOS FUTUROS

Como trabalhos futuros é proposto:

1. Captura de dados em uma aplicação bancária real e *online*, para que o *framework* seja validado em um ambiente totalmente mais próximo do uso final;
2. Preparar uma infraestrutura que aceite a coleta e armazenamento de dados com mais usuários;
3. Aplicar os experimento com um número maior de usuários, na casa das centenas e milhares;
4. Desenvolver o experimento por um período de tempo maior, meses, trimestres ou semestres;
5. Aprofundar o estudo dos algoritmos de Aprendizado de Máquina, com estudos dos parâmetros e investigação das abordagens realizadas;
6. Esgotar a quantidade de características utilizadas nos modelos;
7. Aplicar filtros sobre os dados capturados dos sensores, visando melhorar a performance dos modelos criados;
8. Analisar a relação entre os modelos de celulares e qualidade dos dados de sensores capturados, e reflexo na qualidade do modelo gerado para o usuário;
9. Explorar técnicas de ataques às quais o sistema proposto pode ser suscetível, para propor e implementar contramedidas que tornem o sistema mais resiliente.

REFERÊNCIAS BIBLIOGRÁFICAS

1 2019 Mobile Industry Impact Report: Sustainable Development Goals Executive Summary. 2019. Disponível em: <https://www.gsma.com/betterfuture/2019sdgimpactreport/wp-content/uploads/2019/09/SDG_Report_2019_ExecSummary_Web_Singles.pdf>.

2 NUMBER of smartphone users worldwide from 2016 to 2021. 2020. Disponível em: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.

3 DIGITAL Banking Users to Reach 2 Billion This Year, Representing Nearly 40% of Global Adult Populati. Disponível em: <https://www.juniperresearch.com/press/press-releases/digital-banking-users-to-reach-2-billion>.

4 MOORE, M. *Mobile malware attacks double in 2018*. ITProPortal, 2019. Disponível em: <https://www.itproportal.com/news/mobile-malware-attacks-double-in-2018/>.

5 ALI, M. L.; MONACO, J. V.; TAPPERT, C. C.; QIU, M. Keystroke biometric systems for user authentication. *Journal of Signal Processing Systems*, Springer, v. 86, n. 2-3, p. 175–190, 2017.

6 ALPAR, O. Biometric touchstroke authentication by fuzzy proximity of touch locations. *Future Generation Computer Systems*, Elsevier, v. 86, p. 71–80, 2018.

7 ANTAL, M.; SZABÓ, L. Z.; LÁSZLÓ, I. Keystroke dynamics on android platform. *Procedia Technology*, Elsevier, v. 19, p. 820–826, 2015.

8 FRANK, M.; BIEDERT, R.; MA, E.; MARTINOVIC, I.; SONG, D. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, IEEE, v. 8, n. 1, p. 136–148, 2012.

9 KHAN, H.; ATWATER, A.; HENGARTNER, U. A comparative evaluation of implicit authentication schemes. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2014. p. 255–275.

10 ALARIKI, A. A.; MANAF, A. A.; MOUSAVI, S. M. Features extraction scheme for behavioural biometric authentication in touchscreen mobile devices. *International Journal of Applied Engineering Research*, v. 11, n. 18, p. 9331–9344, 2016.

11 SHEN, C.; LI, Y.; CHEN, Y.; GUAN, X.; MAXION, R. A. Performance analysis of multi-motion sensor behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 13, n. 1, p. 48–62, 2017.

12 TEH, P. S.; ZHANG, N.; TEOH, A. B. J.; CHEN, K. A survey on touch dynamics authentication in mobile devices. *Computers & Security*, Elsevier, v. 59, p. 210–235, 2016.

13 SHIH, D.-H.; LU, C.-M.; SHIH, M.-H. A flick biometric authentication mechanism on mobile devices. In: IEEE. *2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*. [S.l.], 2015. p. 31–33.

14 LAMICHE, I.; BIN, G.; JING, Y.; YU, Z.; HADID, A. A continuous smartphone authentication method based on gait patterns and keystroke dynamics. *Journal of Ambient Intelligence and Humanized Computing*, Springer, v. 10, n. 11, p. 4417–4430, 2019.

15 MAHFOUZ, A.; MAHMOUD, T. M.; ELDIN, A. S. A survey on behavioral biometric authentication on smartphones. *Journal of information security and applications*, Elsevier, v. 37, p. 28–37, 2017.

16 MOTIONEVENT. Disponível em: <https://developer.android.com/reference/android/view/MotionEvent>.

17 SENSOREVENT. Disponível em: <https://developer.android.com/reference/android/hardware/SensorEvent.html>.

18 SENSORES de movimento. Disponível em: <https://developer.android.com/guide/topics/sensors/sensors_motion?hl=pt>.

19 VISãO geral dos sensores. Disponível em: <https://developer.android.com/guide/topics/sensors/sensors_overview>.

20 PUTRI, A. N.; ASNAR, Y. D. W.; AKBAR, S. A continuous fusion authentication for android based on keystroke dynamics and touch gesture. In: IEEE. *2016 International Conference on Data and Software Engineering (ICoDSE)*. [S.l.], 2016. p. 1–6.

21 SAINI, B. S.; KAUR, N.; BHATIA, K. S. Authenticating mobile phone users based on their typing position using keystroke dynamics. In: SPRINGER. *Proceedings of 2nd International Conference on Communication, Computing and Networking*. [S.l.], 2019. p. 25–33.

22 FRIDMAN, L.; WEBER, S.; GREENSTADT, R.; KAM, M. Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. *IEEE Systems Journal*, IEEE, v. 11, n. 2, p. 513–521, 2016.

23 GERóN, D. *Machine Learning: This Book Includes: Machine Learning for Beginners, Machine Learning with Python (English Edition)*. [S.l.]: Amazon, 2019.

24 GRUS, J. *Data Science do Zero: Primeiras Regras com o Python*. 1. ed. [S.l.]: Alta books, 2016. v. 1.

25 MAYNARD, M. *Machine Learning: Introduction to Supervised and Unsupervised Learning Algorithms with Real-World Applications (Advanced Data Analytics Book 1)*. [S.l.]: Amazon, 2020.

26 SAEVANEE, H.; BHATARAKOSOL, P. User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device. In: IEEE. *2008 International Conference on Computer and Electrical Engineering*. [S.l.], 2008. p. 82–86.

27 WU, J.; CHEN, Z. An implicit identity authentication system considering changes of gesture based on keystroke behaviors. *International Journal of Distributed Sensor Networks*, v. 11, n. 6, p. 470274, 2015.

28 ROH, J.-H.; LEE, S.-H.; KIM, S. Keystroke dynamics for authentication in smartphone. *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, 2016.

29 SMITH-CREASEY, M.; ALBALOOSHI, F. A.; RAJARAJAN, M. Context awareness for improved continuous face authentication on mobile devices. *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2018.

30 ALARIKI, A. A.; MANAF, A. A. Touch gesture authentication framework for touch screen mobile devices. *Journal of Theoretical & Applied Information Technology*, v. 62, n. 2, 2014.

31 MARCEL, S. Beat–biometrics evaluation and testing. *Biometric technology today*, Elsevier, v. 2013, n. 1, p. 5–7, 2013.

32 POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Bioinfo Publications, 2011.

33 ZHANG, D.; WANG, J.; ZHAO, X. Estimating the uncertainty of average f1 scores. In: *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. [S.l.: s.n.], 2015. p. 317–320.

34 DOMINGOS, N. d. S. *Biometrias comportamentais em dispositivos móveis*. Tese (Doutorado), 2014.

35 BRODERSEN, K. H.; ONG, C. S.; STEPHAN, K. E.; BUHMANN, J. M. The balanced accuracy and its posterior distribution. In: IEEE. *2010 20th International Conference on Pattern Recognition*. [S.l.], 2010. p. 3121–3124.

36 BURIRO, A.; GUPTA, S.; CRISPO, B. Evaluation of motion-based touch-typing biometrics for online banking. In: IEEE. *2017 international conference of the biometrics special interest group (BIOSIG)*. [S.l.], 2017. p. 1–5.

37 TEMPER, M.; TJOA, S.; KAISER, M. Touch to authenticate—continuous biometric authentication on mobile devices. In: IEEE. *2015 1st International Conference on Software Security and Assurance (ICSSA)*. [S.l.], 2015. p. 30–35.

38 SHILA, D. M.; SRIVASTAVA, K.; ONEILL, P.; REDDY, K.; SRITAPAN, V. A multi-faceted approach to user authentication for mobile devices — using human movement, usage, and location patterns. *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, 2016.

39 DHAGE, S.; KUNDRA, P.; KANCHAN, A.; KAP, P. Mobile authentication using keystroke dynamics. In: IEEE. *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*. [S.l.], 2015. p. 1–5.

40 ESTRELA PRISCILA MORAIS ARGôLO BONFIM, A. D. M. e. A. R. d. O. e. G. W. F. e. A.-N. G. D. e. N. A. S. Estudo experimental da biometria comportamental para autenticação contínua deusuários em aplicações bancáriasmobile. IADIS, 2019.

41 ESTRELA PRISCILA MORAIS ARGôLO BONFIM, A. D. M. e. A. R. d. O. e. G. W. F. e. A.-N. G. D. e. M. F. L. L. d. Biotouch: a framework based on behavioral biometrics and location for continuous authentication on mobile banking applications. IEEE, 2020.

42 FIREBASEINSTANCEID. Disponível em: <https://firebase.google.com/docs/reference/android/com/google/firebase/iid/FirebaseInstanceId>.

43 FIREBASE. Disponível em: <https://firebase.google.com/>.

44 HARRISON, M. *Machine Learning – Guia de Referência Rápida: Trabalhando com dados estruturados em Python*. 1. ed. [S.l.]: O'Reilly, Novatec, 2020. v. 1. (1, v. 1). ISBN 9788575228180.

45 MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997.

46 RUSSELL, S. J.; NORVIG, P. *Inteligncia Artificial*. 3. ed. [S.l.]: Elsevier, 2013. v. 1.

47 AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Apress, 2015.

48 1.4. Support Vector Machines. Disponível em: <https://scikit-learn.org/stable/modules/svm.html#svm>.

49 SKLEARN.SVM.SVC. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>.

50 BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

51 SMITH, C. *Decision trees and random forests: a visual introduction for beginners*. [S.l.]: Blue Windmill Media, 2017.

52 BÜHLMANN, P. Bagging, boosting and ensemble methods. In: *Handbook of computational statistics*. [S.l.]: Springer, 2012. p. 985–1022.

53 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html?highlight=randomforestclassifier#sklearn.ensemble.RandomForestClassifier>.

54 LUSA, L. et al. Gradient boosting for high-dimensional prediction of rare events. *Computational Statistics & Data Analysis*, Elsevier, v. 113, p. 19–37, 2017.

55 NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, Frontiers, v. 7, p. 21, 2013.

56 3.2.4.3.5. sklearn.ensemble.GradientBoostingClassifier. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html?highlight=gradient\%20boosting#sklearn.ensemble.GradientBoostingClassifier>.

57 CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794.

58 XGBOOST Parameters. Disponível em: <https://xgboost.readthedocs.io/en/latest/parameter.html>.

59 KHAN, H.; HENGARTNER, U.; VOGEL, D. Mimicry attacks on smartphone keystroke authentication. *ACM Transactions on Privacy and Security (TOPS)*, ACM New York, NY, USA, v. 23, n. 1, p. 1–34, 2020.

60 STANCIU, V.-D.; SPOLAOR, R.; CONTI, M.; GIUFFRIDA, C. On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. In: *proceedings of the sixth ACM conference on data and application security and privacy*. [S.l.: s.n.], 2016. p. 105–112.