

MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA

Victor Villas Bôas Chaves  
Lucas Sousa Meireles  
Claudio Cavalcante Bomfim Júnior

# Agentes Inteligentes no Reconhecimento de Usuários de Sistemas Computacionais Baseado na Dinâmica de Digitação

Rio de Janeiro  
Outubro 2017

Instituto Militar de Engenharia

Victor Villas Bôas Chaves

Agentes Inteligentes no Reconhecimento de Usuários de Sistemas Computacionais Baseado na  
Dinâmica de Digitação

Projeto de Final de Curso apresentado ao Curso de Engenharia de Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Engenheiro de Computação.  
Orientador: Prof. Ronaldo Goldschmidt - D.C.

Rio de Janeiro  
Outubro 2017

cOutubro 2017

INSTITUTO MILITAR DE ENGENHARIA  
Praça General Tibúrcio, 80 - Praia Vermelha  
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmар ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

/ Victor Villas Bôas Chaves; orientados por Prof. Ronaldo  
Goldschmidt- Rio de Janeiro: Instituto Militar de Engenharia, Outubro  
2017.

18p. : il.

- Instituto Militar de Engenharia - Rio de Janeiro, Outubro 2017.

# Resumo

**Palavras-chaves:** dinâmica da digitação, reconhecimento de usuário

# Abstract

**Keywords:** keyboard dynamics, user authentication

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	Contexto e Motivação	7
1.2	Objetivos	7
1.3	Contribuições Esperadas	8
1.4	Método	8
1.5	Viabilidade	8
1.6	Cronograma	8
1.7	Organização do Texto	9
<b>2</b>	<b>FUNDAMENTAÇÃO</b>	<b>10</b>
2.1	Problema de Classificação	10
2.2	Erro de amostragem	10
2.3	Dimensão de Vapnik-Charvanenks ( $D_{vc}$ )	12
2.4	Validação	12
2.5	Trabalhos Relacionados	13
<b>3</b>	<b>SOLUÇÃO PROPOSTA</b>	<b>14</b>
3.1	Infraestrutura	14
3.1.1	Redes	15
3.1.2	Máquinas Virtuais	15
3.1.3	Dados	15
3.1.4	Processamento	16
3.2	Modelo conceitual	16
3.3	Protótipo	16
<b>4</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>17</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>18</b>
	<b>REFERÊNCIAS</b>	<b>19</b>

# 1 Introdução

## 1.1 Contexto e Motivação

Autenticação é o processo de verificar positivamente a identidade de um usuário, dispositivo ou outra entidade em um sistema computacional, em geral como prerequisite de permissão de acesso a recursos. A autenticação de uma entidade alcança verificação positiva ao corretamente corresponder uma forma de identificação (como uma chave secreta) previamente definida. (O’GORMAN, 2003)

É conveniente discernir a diferença entre autenticação de dispositivos (máquinas) e usuários (pessoas), sendo o último sistema onde concentram-se diversas dificuldades em função da usabilidade e interface para o usuário. (O’GORMAN, 2003) O uso de senhas que humanos possam lembrar é por si só um fator de risco, havendo portanto a necessidade de introduzir mecanismos secundários de autenticação que não introduzam queda de usabilidade significativa (MORRIS; THOMPSON, 1979).

Dentre as alternativas de autenticação multifator, o uso de características biométricas se destaca por sua natureza individual e difícil falsificação. Utilizados principalmente em sistemas de autorização e autenticação em ambientes físicos, um sistema biométrico se baseia em um sensor consistente de assinatura biométrica (Ed Dawson et al., 2003).

As características biométricas frequentemente mencionadas são as fisiológicas, mas seu emprego traz diversos fatores complicantes como a necessidade de amostragem prévia e diminuição da usabilidade do sistema. Uma alternativa é o uso de características biométricas de comportamento, como padrões comportamentais expressos naturalmente pelo usuário (MOSKOVITCH et al., 2009).

As vantagens do uso de características comportamentais incluem a possibilidade de amostragem silenciosa, maior variabilidade do grau de confiança e a transparência do mecanismo para o usuário, o campo da segurança da informação mantém-se sem muitos meios de autenticação neste sentido. Em particular, sistemas providos pela *Web* em geral possuem uma uniformidade de interface que permite a coleta de vários padrões comportamentais durante todo o uso do sistema (MOSKOVITCH et al., 2009).

## 1.2 Objetivos

Esse trabalho tem como objetivo geral a construção de um sistema de autenticação baseado em dinâmica de digitação do usuário, disponibilizado como serviço externo interagindo através de uma API, além de desenvolver uma biblioteca que facilite sua integração a outros componentes de serviços Web, e conta como objetivos específicos:

- Analisar os tipos de informação que se podem extrair a partir dos padrões de digitação de um indivíduo;
- Modelar a combinação das informações extraídas utilizando algoritmos de aprendizado de máquina;
- Sistematizar um mecanismo de coleta de amostras que permita o treinamento dos modelos escolhidos;
- Definir uma arquitetura de sistema para implantação dos mecanismos de coleta e autenticação definidos;

### 1.3 Contribuições Esperadas

Espera-se desse projeto a produção de um mecanismo de autenticação com total transparência ao usuário, como um serviço complementar de autenticação do mesmo, com a finalidade de impedir fraudes de acesso as aplicações do cliente, sem que o usuário esteja ciente da aplicação. Deste modo o agente malicioso não tomará conhecimento deste segundo método de autenticação e não estará apto a se adaptar ao padrão de digitação de quem este pretende invadir, o projeto visa assim suprir a necessidade de autenticação transparente da qual carece o mercado de segurança da informação, por meio de padrões compartimentais, os quais são mais difíceis de se simular.

### 1.4 Método

O desenvolvimento do serviço de autenticação e da biblioteca será feito de forma paralela. Por um lado a biblioteca é responsável por coletar os dados de dinâmica de digitação no navegador de um usuário, enquanto o serviço receberá esses dados e fará a verificação de identidade.

Após a arquitetura do sistema estar definida, serão experimentados modelos de aprendizado de máquina em um serviço protótipo para prova de conceito. Durante esse experimento poderá ser feita coleta de dados e ao mesmo tempo avaliar a usabilidade da biblioteca.

### 1.5 Viabilidade

O desenvolvimento da biblioteca de interação com a API, por ser um produto independente do desenvolvimento do sistema de autenticação, pode ser ajustado conforme a disponibilidade de tempo do projeto. Portanto, a viabilidade do projeto se dá principalmente como dependente da prototipagem e experimentação do sistema.

Como o IME possui diversas oportunidades de experimentação com seus alunos em ambiente EAD, o experimento é factível e pode ser feito nos moldes de trabalhos semelhantes anteriores. Em todo caso, há *datasets* públicos de dinâmica de digitação que podem ser utilizados caso o desenvolvimento do sistema ultrapasse o cronograma esperado e inviabilize a coleta de dados local.

### 1.6 Cronograma

	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out
Revisão bibliográfica									
Modelagem conceitual									
Prototipagem									
Coleta de dados									
Entrega de relatório parcial									
Experimentação									
Entrega do relatório final									
Apresentação									

Tabela 1 – Cronograma mensal de trabalho



Definição da arquitetura básica do produto	VE
Análise de trabalhos relacionados ao tema proposto	VC
Prototipagem do serviço de autenticação	VC
Escolha dos modelos de aprendizado de máquina	VC
Esquematização da API e modelagem do sistema	VC
Experimentação do serviço via API em um ambiente simulado	VF
Análise dos resultados e relatório da efetividade do sistema	VF

Tabela 2 – Tarefas entregáveis

## 1.7 Organização do Texto

No capítulo 2.4 são introduzidos os conceitos necessários para a modelagem conceitual de um sistema de autenticação por dinâmica de digitação. Na seção 2.5 são discutidos trabalhos relacionados.

No capítulo 3 é apresentado uma arquitetura de sistema de autenticação isolado, para fácil implantação do método apresentado. Na seção 3.2 é definido o modelo de autenticação, especificando o fluxo de informações desde a coleta até a decisão de um grau de confiança de identidade, enquanto em 3.3 é demonstrada uma possível implementação da solução proposta, servindo como prova de conceito para o modelo.

No capítulo 4 são analisados os resultados do experimento proposto com o protótipo criado, analisando o sucesso da solução.

No capítulo 5 termina-se por sumarizar o conceito, a solução e os resultados obtidos pelo sistema apresentado.

## 2 Fundamentação

O problema que o ramo de aprendizado por máquinas se propõe a resolver é a busca pela aproximação funcional matemática de um problema real em um conjunto matemático, onde  $F : C \rightarrow R$  é a função alvo desconhecida, então para resolver o problema é suposto uma função  $G : H \rightarrow R$ , onde  $G$  é a função estimada que pretende-se aproximar de  $F$ ,  $H \subset C$  é o conjunto das amostras que pretende-se expandir para o conjunto real expandido  $C$ , os valores de  $H$  e  $R$  são conhecidos; porém, para um problema não interpolável, não há um mapeamento conhecido de  $F : C \rightarrow R$ , seja por falta parâmetros de difícil análise, ou pelo caráter indeterminado do problema, em ambos os casos o tratamento é similar através de estimação e tratamento de erros na abordagem.

### 2.1 Problema de Classificação

O problema de traduzir um conjunto amostral  $H$  em conjuntos distintos com uma abordagem não interpolável é conhecido como o problema classificação, para tal pode se usar, no domínio discreto, a seguinte abordagem. Seja  $f : x \rightarrow y$  a função alvo onde  $x = [x_1, x_2, \dots, x_d]^t$  é o vetor das amostras de  $H$ , seja o somatório  $\sum_{i=1}^d \omega_i * x_i$ , onde  $\omega_i$  é o peso do seu respectivo elemento do vetor  $x$ , de tal forma que:

$$\sum_{i=1}^d \omega_i * x_i > t; y = 1. \quad \sum_{i=1}^d \omega_i * x_i < t; y = 0. \quad (2.1)$$

De tal forma os pesos tem suas características definidas por seus respectivos elementos de  $x$  da seguinte forma:

$|\omega_i|$  é alto quando  $x_i$  for importante.

$\omega > 0$  quando  $x_i$  for benéfico.

$\omega < 0$  quando  $x_i$  for maléfico.

Por fim teremos uma aproximação da função alvo no problema de classificação para  $h(x) = \text{sign}((\sum_{i=1}^d \omega_i * x_i) + \omega_0)$ , onde  $\omega_0 = t$ .

Neste caso o problema de classificação se restringirá a apenas dois conjuntos alvo em  $R$ , para uma generalização com múltiplos conjuntos usam-se estratégias de modo a recair no problema com dois conjuntos alvo.

### 2.2 Erro de amostragem

Vale notar que o vetor  $x$  apenas representa as amostras colhidas, assim embora ele represente o conjunto  $C$  como uma aproximação de  $C$  para  $H$  existe uma divergência conhecida como erro dentro da amostra  $E_{in}$  essa diferença é representada pela diferença entre  $x$  e sua projeção em  $C$ , assim como a extrapolação da função nem sempre vai atender o comportamento esperado pela função estimada  $G : H \rightarrow R$  assim a divergência entre o real e o esperado é o chamado fora da amostra  $E_{out}$ , o qual funciona como um meio de verificar o quão errada esta a estimação de  $G : H \rightarrow R$ .

Em um problema de aprendizado a verificação de erros não é um problema trivial, uma vez que pela impossibilidade de se conferir ambos os erros,  $E_{in}$  e  $E_{out}$ , de maneira direta, por não conhecermos a função

alvo, usam-se ferramentas que possibilitam verificar tais erros e assim modificar de maneira eficiente o método de estimação da função, uma primeira ferramenta usada para verificação de erros é a desigualdade de Hoeffding Chernoff como meio de verificação sobre os erros dentro e fora da amostra, a desigualdade em questão pode ser expressa da seguinte forma:

$$\begin{aligned} P[|v - \mu| > \epsilon] &\leq 2 * e^{-2 * \epsilon^2 * N}, \epsilon > 0 \\ P[|v - \mu| \leq \epsilon] &> 2 * e^{-2 * \epsilon^2 * N}, \epsilon > 0 \end{aligned} \quad (2.2)$$

A fim de melhor adaptar os valores, leia-se  $P[x]$  como probabilidade de  $x$ , e diminuir os erros escolhe-se o valor de  $\epsilon$  de tal forma que  $v + \epsilon \geq \mu \geq v - \epsilon$ , nota-se que a fronteira denotada por  $2 * e^{-2 * \epsilon^2 * N}$  não depende de  $\mu$ , nem do tamanho do domínio.

Adaptando-se a desigualdade ao problema de classificação ela é reescrita de forma que  $|E_{in} - E_{out}| = |v - \mu|$  assim tem-se:

$$\begin{aligned} P[|E_{in} - E_{out}| > \epsilon] &\leq 2 * |H| * e^{-2 * \epsilon^2 * N} \\ P[|E_{in} - E_{out}| \leq \epsilon] &> 2 * |H| * e^{-2 * \epsilon^2 * N}, \epsilon > 0 \end{aligned} \quad (2.3)$$

Como o erro  $E_{in}$  é uma medida que advem da diferença entre a projeção da função estimada pode há uma identidade para tal, na qual o  $E_{in}$  é a média das distâncias entre o valor real e o estimado, os seja:

$$\begin{aligned} E_{in} &= \frac{\sum_{i=0}^N L(g(x_i), f(x_i))}{N} \\ E_{in} &= \frac{\sum_{i=0}^N L(g(x_i), y_i)}{N} \end{aligned} \quad (2.4)$$

Quando a função alvo é indeterminada, dessa forma como o  $E_{in}$  é independente de  $E_{out}$  além de ser identicamente distribuída a tal pode-se afirmar que o valor de  $E_{out}$  converge para  $E_{out} = E(E_{in}) = E(\frac{\sum_{i=0}^N L(g(x_i), y_i)}{N})$ , leia  $E(x)$  como a esperança de  $x$ , logo o erro de classificação é representado como  $P[g(x) \neq y] = E(L(g(x), y))$  de tal forma para minimizar de maneira mais eficiente se estipula parametros afim de diminuir o erro quadrático médio assim buscam-se os parâmetros, tais que  $(x, y) \rightarrow \inf E((y - g(x))^2)$ , leia-se  $\inf$  como infimo.

Afim de otimizar a estimação busca-se então um classificador ótimo ao problema tal que o  $E_{out}$  seja mínimo sem para isso alterar a amostra, de tal forma  $E_{out}^* = \inf(E_{out}(h))$ ,  $h \in H$  nesse caso o erro fora da amostra é minimizado quando ele se aproxima do erro para um classificador ótimo  $\inf(E_{out} - E_{out}^*)$ , seja  $\bar{h} \in H$ , tal que  $\bar{h}$  é a amostra ideal, nesse caso  $E_{out} - E_{out}^* = (E_{out} - E_{out}(\bar{h})) + (E_{out}(\bar{h}) - E_{out}^*)$ , onde  $E_{out} - E_{out}(\bar{h})$  é o erro na estimação da amostra e  $E_{out}(\bar{h}) - E_{out}^*$  o erro na aproximação da amostra, assim minimizar o erro de estimação implica em aumentar o erro de aproximação, bem como assim minimizar o erro de aproximação implica em aumentar o erro de estimação, no entanto o erro de estimação esta diretamente ligado ao  $E_{in}$ , portanto minimizar o  $E_{in}$  e consequentemente o  $E_{out}$  implicam em aumento do erro de aproximação, logo existe um risco associado quando se tenta adaptar a função estimada ao comportamento real, o que pode levar ao problema de overfitting, para impedi-lo avalia-se então o risco associado a relação entre os erros conforme aa seguinte forma:

$$E_{out} - E_{out}^* \leq E_{out} - E_{in} - E_{out}^* + E_{in}^* \leq \text{Sup}(|E_{out} - E_{in}|) \quad (2.5)$$

Assim usando-se da desigualdade (3) para isolar o valor de  $\epsilon$ , tem-se que o valor de  $Sup(|E_{out} - E_{in}|)$  substituindo o valor  $\epsilon$  na equação resultante chega-se a:

$$Sup(|E_{out} - E_{in}|) \leq \alpha \sqrt{\frac{\log(|H|)}{N\delta}} \quad (2.6)$$

Dessa forma é possível limitar a variação máxima entre os dois erros afim de minimizar o risco associado ao overfitting, para usar essa relação de maneira formal no ambito do problema ainda é preciso analisar as caraterísticas do conjunto  $H$ , de onde são colhidas as amostras para teste e treino, para tal busca-se a divisão em subconjuntos de maneira que estes sejam disjuntos e  $H$  é então separado como um conjunto, tal que  $H(s) = \{h(x_1), h(x_2), \dots, h(x_N)\}$ , com  $s \in X^N$ , assim o crescimento de  $H$  será delimitado por  $2^N$  e consequentemente o seu número de dicotomias ( $m_h(H)$ ), que são as formas de separar  $H$  em duas partes, será a forma de se avaliar o crescimento de  $H$  afim de reduzir os riscos ao selecionar a abordagem e tratar os erros do problema de classificação.

### 2.3 Dimensão de Vapnik-Charvanenks ( $D_{vc}$ )

Ainda assim é necessária uma ferramenta que defina qual o maior subconjunto de  $H$ , o qual possa ser separado dicotomicamente em  $H$ , pois a análise anterior foi feita integralmente com base no supremo da diferença entre os erros em  $H$  para tal foi criada a dimensão de Vapnik-Charvanenks, assim:

$$\begin{aligned} d_{vc}(H) &= \max\{N : m_H\} \\ d_{vc}(H) &= \infty, \text{ se } (m_h(N) = 2^N, \forall N) \end{aligned} \quad (2.7)$$

a dimensão VC em sí é uma ferramenta de difícil estimação, para facilitar a sua utilização em vista da minimização de risco é utilizado o Lema de Sauer, tomando  $d_{vc} = d$ , implicando em:

$$m_h(N) \leq \sum_{i=0}^d \binom{N}{i} = \phi_d(N) \leq \left(\frac{eN}{d}\right)^d \quad (2.8)$$

Ainda assim o ajuste ainda estará sujeito a erros de generalização quando a tese escolhida  $G : H \rightarrow C$ , quando for colocado em prática, tal erro é exposto como erro de generalização denotado como  $argMin E_{in}(h) = arg_{h \in H} \frac{1}{N} \sum L(h(x), y)$ , leia  $Min$  como mínimo, a fim de minimizar esses erros bem como seus riscos tornam-se necessárias táticas de verificação e validação da tese.

### 2.4 Validação

A validação é uma técnica que visa tornar a tática escolhida em otimal, assim sendo a validação é principalmente um meio de garantir que a abordagem escolhida para o problema esteja tão próxima da realidade quanto for possível, neste caso o conjunto de amostras é tratado para aprimorar a aplicação da tática, o tratamento pode funcionar de maneiras de ferentes de uma tática para outra, por fim a tática de validação mais utilizada e precisa quanto a validar o modelo é a validação cruzada, a qual apesar da eficiência e precisão pode não ser utilizada em detrimento de outras táticas por ser muito custosa em termos computacionais.

A validação cruzada se baseia em redistribuir a amostra de teste em diversas sub-amostras de treino para então escolher-se recursivamente tais conjuntos de amostras e usar o restante como amostra de treino esta abordagem visa estimar o valor do  $E_{out}$  que não pode ser calculado sem treino ou verificação do modelo já

em uso, então calculam-se os erros para assim se escolher como amostra de teste a amostra, a qual contenha os melhores índices de erro; porém, como visto previamente, forçar os valores de erro na amostra e fora dela para minimiza-los pode acarretar em problemas de overfitting e neste caso os erros calculados durante a validação iriam distuar drasticamente do seu valor real.

Outra técnica de validação amplamente utilizada é a reamostragem, nesse caso o modelo escolhido, mesmo após processos de verificação, não atende as expectativas quando implementado e por isso necessita-se refazer o treinamento, assim sendo se colhem novas amostras para refazer o processo de treino, é importante que não se usem as amostras já colhidas, pois estas iriam moldar o novo modelo com certo juízo de valor, contribuindo fatalmente para o problema de overfitting.

Outra forma de validar um modelo é pela verificação continua durante o treino, neste caso a cada iteração do treinamento calculam-se os erros do modelo em relação ao esperado para assim melhorar o modelo com alterações pontuais, similar ao processo de entrega de um produto em que a cada versão busca-se melhorá-lo e consertar os problemas descobertos, assim como a tática de validação cruzada esse método é muito custoso computacionalmente e por isso pode ser desestimulado em relação aos outros.

## 2.5 Trabalhos Relacionados

Afim de prover ao modelo com características de permisibilidade ao meio de implementação, ou seja, evitar a abordagem de autenticação por outros meios biométricos, pois estes podem implicar na necessidade de equipamentos adicionais, como as técnicas de autenticação da iris(BOWYER; FLYNN, 2013), ou da face(PARKHI, 2015), ou da voz(PADMANABHAN; PREMKUMAR, 2015), a introdução do mecanismo aos moldes de autenticação biométrica se torna possível através da dinamica de digitação, cuja abordagem se aproxima da usada por token rings, uma opção indesejável em certas situações pela sua capacidade de interromper processos importantes.

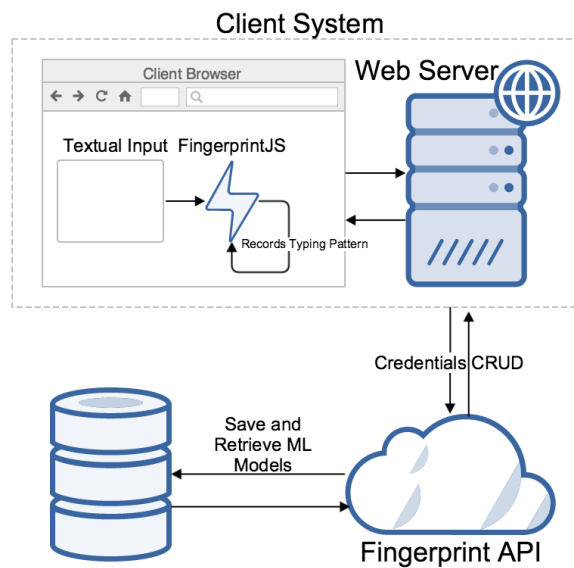
As características mais importantes da aplicação serão a capacidade multi-plataforma e a transparência total ao usuário(WOODARD, 2012), assim avaliando a veracidade do usuário sem que esse tome conhecimento desse processo, como o sistema desejado pretende ser permissível, multi-plataforma, para isso o principal problema dessa forma de autenticação, o qual seria a ocorrência de interrupções indesejáveis, passa a ser do controlador da plataforma de implementação.

## 3 Solução Proposta

O sistema de autenticação proposto consiste em um serviço que ofereça as operações básicas de manutenção de credenciais (*CRUD*) para usuários de clientes mais uma operação de verificação. Essas funcionalidades são acessadas via API REST, a critério do sistema que requer autenticação de seus usuários.

A interface de verificação exigirá a identificação de um usuário e os dados de dinâmica de digitação a serem analisados e responderá a requisição com um indicador de aceitação da identidade. Tal indicador pode ser binário ou assumir valores contínuos, a depender do modelo utilizado. Como utilizar a aceitação do padrão digitação como critério de autenticação fica a critério do cliente, que pode admitir diferentes níveis de tolerância ou diferentes ações de resposta a uma verificação negativa.

Para auxiliar o uso da API, a biblioteca desenvolvida automatizará a coleta de dados de dinâmica de digitação nos campos definidos pelo cliente em seu site, além de disponibilizar métodos de tratamento de dados coletados como truncamento e sub-amostragem. Este componente pode ser deixado em código aberto, já que apenas facilita a interação com a API.



### 3.1 Infraestrutura

Embora o serviço conste de uma API - podendo ser, portanto, implantado como um único servidor e posteriormente escalado por replicação, deseja-se definir também a infraestrutura de processamento de dados. Para o funcionamento do serviço é necessário que modelos preditivos sejam constantemente retreinados de forma automatizada.

Como a carga de um sistema de processamento offline é extremamente variável, buscamos uma arquitetura de servidores e serviços que seja escalável e flexível com baixo custo. O provedor de computação em nuvem escolhido foi a Amazon Web Services devido a sua grande flexibilidade e baixo custo.

O modelo esquemático do sistema de processamento está representado na figura 3.1.

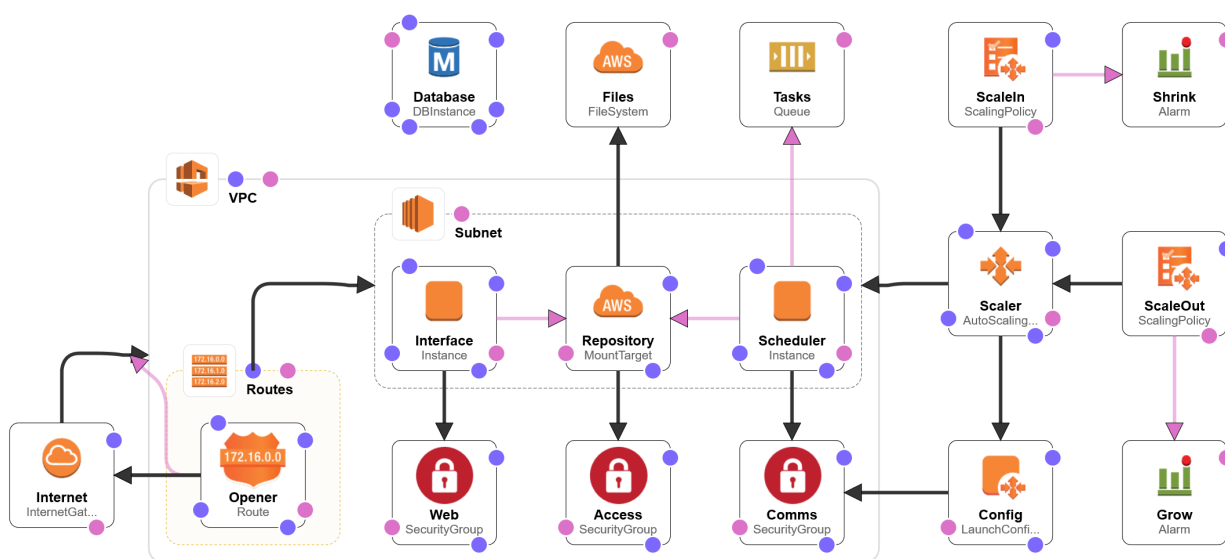


Figura 1 – Diagrama de serviços no *Cloud Formation*.

### 3.1.1 Redes

No modelo de nuvem da Amazon Web Services (AWS), as máquinas virtuais precisam estar dentro de uma Rede Virtual Privada (**VPC** - Virtual Private Network), que é dividida em Subredes (**Subnet**) especificadas em diferentes Zonas de Disponibilidade (Availability Zones, representam os diferentes servidores da AWS pelo mundo).

Para dar acesso à Internet para as máquinas, é necessário associar um Gateway de Internet (*Internet* - Internet Gateway) à **VPC** e uma tabela de rotas de roteamento (**Routes**) contendo uma rota de abertura (**Opener** - Route) para a **Subnet** para redirecionar o tráfego ao endereço local do gateway.

### 3.1.2 Máquinas Virtuais

Duas máquinas virtuais persistentes serão utilizadas para o sistema de processamento. Uma delas agindo como servidor *web* para a interface gráfica de controle e monitoramento do sistema (**Interface**), enquanto a outra conterá o serviço de agendamento de tarefas, agindo como nó mestre do cluster de processamento (**Scheduler**).

Para facilitar a coordenação da interface, do agendador e dos nós de processamento, todas as máquinas irão compartilhar dados por um volume montado em comum em seus sistemas de arquivos (**Repository** - Mount Target). Esse volume montado está associado a um Sistema de Arquivos Elástico (**Files** - Elastic File System).

### 3.1.3 Dados

Além do sistema de arquivos, é necessário um banco de dados para persistir as informações relativas ao agendamento de tarefas além dos resultados do processamento (**Database** - DbInstance). Em particular, escolhemos utilizar o PostgreSQL como SGBD para os dados relacionados à API e para os dados de análise.

Ainda, para poder coordenar os nós de processamento, o **Scheduler** precisará de um mecanismo de comunicação distribuída. Para isso usa-se um serviço de fila de mensagens (**Tasks** - Queue). O sistema de

escalonamento de nós processadores irá utilizar esta fila como parâmetro de decisão: uma fila muito grande de tarefas deverá indicar a necessidade de instanciar mais uma máquina de processamento.

#### 3.1.4 Processamento

As máquinas de processamento serão criadas com base nos gatilhos (**Srhink** e **Grow** - Alarm), que a cada minuto irão verificar o tamanho da fila **Tasks**. Ao sair dos limites estipulados, os alarmes irão acionar respectivamente as políticas de escalonamento (**ScaleOut** e **ScaleIn** - ScalingPolicy) que alocam e desalocam uma máquina de processamento.

O sistema de escalonamento (**Scaler** - AutoScalingGroup) irá instanciar as máquinas dentro da **Subnet** com as especificações de trabalho semelhantes ao nó mestre **Scheduler**, consumindo porém máquinas virtuais do banco de máquinas leiloadas sob demanda (Spot Instances) com menor custo.

### 3.2 Modelo conceitual

### 3.3 Protótipo



## 4 Experimentos e Resultados

## 5 Conclusão

## Referências

- BOWYER, K. P. H. K. W.; FLYNN, P. J. A survey of iris biometrics research: 2008–2010. p. 15–54, 2013.
- Ed Dawson et al. BAAI: biometric authentication and authorization infrastructure. In: *International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003*. IEEE, 2003. p. 274–278. ISBN 0-7803-7724-9. Disponível em: <<http://ieeexplore.ieee.org/document/1270620/>>.
- MORRIS, R.; THOMPSON, K. Password security: a case history. *Communications of the ACM*, ACM, v. 22, n. 11, p. 594–597, nov 1979. ISSN 00010782. Disponível em: <<http://portal.acm.org/citation.cfm?doid=359168.359172>>.
- MOSKOVITCH, R. et al. Identity theft, computers and behavioral biometrics. In: *2009 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2009. p. 155–160. ISBN 978-1-4244-4172-3. Disponível em: <<http://ieeexplore.ieee.org/document/5137288/>>.
- O’GORMAN, L. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, v. 91, n. 12, p. 2021–2040, dec 2003. ISSN 0018-9219. Disponível em: <<http://ieeexplore.ieee.org/document/1246384/>>.
- PADMANABHAN, J.; PREMKUMAR, M. J. J. Machine learning in automatic speech recognition: A survey. v. 32, p. 1–12, 02 2015.
- PARKHI, A. V. e. A. Z. O. M. Deep face recognition. p. 1–6, 2015.
- WOODARD, S. P. B. e D. L. Biometric authentication and identification using keystroke dynamics: A survey. *journal of pattern recognition research*. p. 116–122, 2012.