

# **Detecção de estresse em tempo real usando dispositivos IoT e aprendizado profundo**

**Gabriel Fernandes, Luiz Nazareth, Rubia Viol, Victor Ströele**

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Juiz de Fora (UFJF)  
Caixa Postal 20.010 – 36.036-900 – Juiz de Fora – MG – Brazil

{gabrielfernandes.silva, luiz.nazareth, rubia.viol}@estudante.ufjf.br

victor.stroele@ufjf.br

**Abstract.** Chronic diseases, such as cardiovascular disease, diabetes, and cancer, have daily stress as a relevant factor in their progression. This study proposes a deep learning architecture to detect real-time stress episodes. Data from IoT devices on the individual's heart rate and location was used to do this. A mobile system processes this data, identifies stress patterns, and sends real-time notifications to the user. The results show that with the approach developed, it is possible to detect episodes of stress, providing real-time alerts to the user.

**Resumo.** Doenças crônicas, como cardiovasculares, diabetes e câncer, têm o estresse diário como fator relevante em sua progressão. Este estudo propõe uma arquitetura com aprendizado profundo para detectar episódios de estresse em tempo real. Para tal, dados de dispositivos IoT sobre frequência cardíaca e localização do indivíduo foram utilizados. Um sistema móvel processa esses dados, identifica padrões de estresse e envia notificações em tempo real para o usuário. Os resultados demonstram que é possível detectar episódios de estresse, fornecendo alertas em tempo real ao usuário.

## **1. Introdução**

O estresse é um fenômeno associado a situações imprevisíveis e incontroláveis, onde o indivíduo percebe uma ameaça ou perigo. Como destacado em [Bauer 2002], essa condição desencadeia respostas fisiológicas, como elevação da frequência cardíaca e liberação de adrenalina, que podem persistir mesmo após a remoção do estímulo estressor. Essas reações, quando recorrentes, estão ligadas ao desenvolvimento de doenças crônicas, como diabetes e cardiopatias [Dieleman et al. 2020].

Com o avanço da tecnologia IoT e de dispositivos vestíveis (*wearables*) é possível a coleta constante de dados biométricos, como a frequência cardíaca, e dados contextuais, como a localização do usuário, mesmo sem a necessidade de conexão à internet. Estudos mostram que a análise integrada desses dados utilizando modelos de aprendizado profundo (*deep learning*) pode identificar padrões de comportamento e detectar alterações que indicam episódios de estresse [Gedam and Paul 2021].

Este estudo propõe uma arquitetura para o desenvolvimento de um sistema *offline-first* baseado em operações de aprendizado de máquina para analisar dados de sensores (como frequência cardíaca) e a geolocalização do usuário em tempo real. O sistema

estuda o comportamento basal do usuário e identifica momentos em que há alterações bruscas, como picos de batimentos cardíacos em locais possivelmente estressantes. A implementação inclui um aplicativo Android que se conecta a dispositivos IoT, armazena dados localmente, como em smartbands, e sincroniza com um servidor em nuvem apenas quando há conectividade, protegendo a privacidade e evitando perda de dados. O sistema gera notificações em tempo real quando detecta instabilidades fisiológicas, alinhando-se a estratégias de prevenção primária.

As principais contribuições deste trabalho incluem a otimização de recursos computacionais em dispositivos móveis, a redução de falsos positivos na detecção de estresse, e a integração de dados heterogêneos, como batimentos cardíacos e coordenadas GPS. A localização é crucial para a análise, visto que dados geográficos como altitude, podem causar uma variação do comportamento de batimentos cardíacos da pessoa.

Além desta introdução, este artigo está organizado da seguinte forma: a Seção 2 traz os trabalhos relacionados a essa pesquisa, a Seção 3 detalha os materiais e métodos utilizados no estudo, e a Seção 4 apresenta os resultados obtidos. Por fim, a Seção 5 discute as conclusões do trabalho, além de sugerir direções para pesquisas futuras.

## 2. Trabalhos Relacionados

Para contextualizar a pesquisa, realizamos uma análise de estudos relacionados à detecção de estresse utilizando tecnologias vestíveis e técnicas de aprendizado de máquina.

Em [Can et al. 2020], foi proposto um sistema de detecção de estresse usando *smartbands*, combinando dados fisiológicos (EDA, HRV) e contexto (clima, atividades). Entretanto, a pesquisa apresenta autorrelatos subjetivos e tamanho reduzido de amostra (16 participantes), além de não explorar técnicas de *deep learning* para séries temporais.

Em [Zhao et al. 2023], os autores propuseram um sistema de detecção de estresse baseado em fusão multimodal (ECG e EDA) e análise multitemporal, combinando características *handcrafted* e *deep learning*. A abordagem alcançou alta precisão (90,05%) no dataset WESAD, superando métodos baseados em fontes únicas. No entanto, o estudo foi limitado por sua dependência de ambientes controlados e pela ausência de avaliação em cenários do mundo real com conectividade intermitente, além de não explorar a personalização contínua dos modelos para usuários individuais.

Em [Risch et al. 2022] foi desenvolvido um algoritmo de LSTM (rede neural recorrente) para detectar COVID-19 dois dias antes do surgimento de sintomas, usando pulseiras que monitoram frequência respiratória, cardíaca e temperatura da pele. O sistema alcançou 68% de sensibilidade em 66 pacientes sintomáticos, mas foi limitado por falsos positivos e falta de diversidade na amostra (apenas adultos saudáveis  $\leq 51$  anos).

O estudo de [Nurmí and Lohan 2021] analisou 67 pesquisas sobre monitoramento de saúde com aprendizado de máquina, explorando sensores (acelerômetros, EEG, ECG e *smartwatches*). Os dispositivos acompanham sinais vitais de pacientes com doenças crônicas, como problemas cardíacos e diabetes. Os resultados concluíram que o uso de sensores corporais é uma abordagem viável para a coleta de dados de saúde.

Este trabalho se distingue das abordagens anteriores pelos seguintes aspectos:

- **Operação autônoma:** Funcionamento independente de conexão contínua à internet, com armazenamento local dos dados no dispositivo móvel;

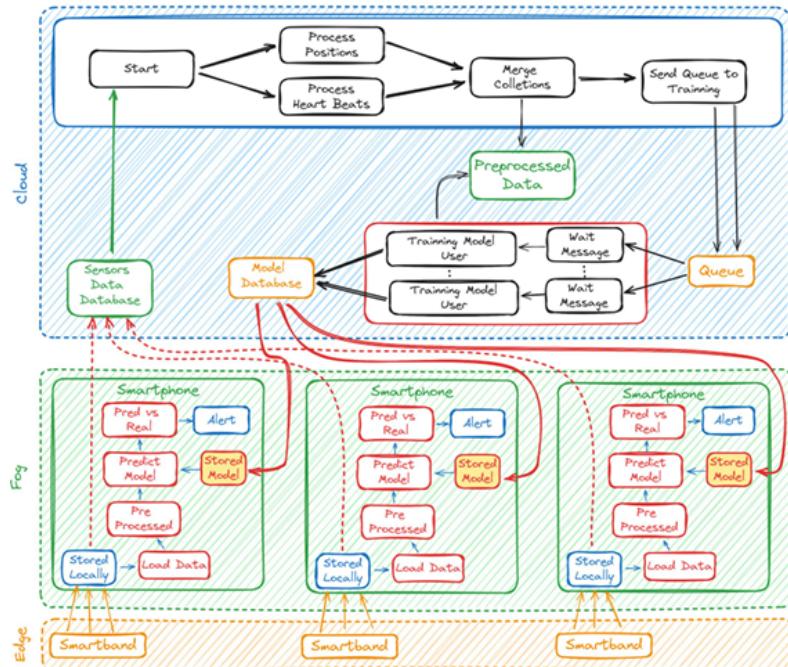
- **Aprendizado adaptativo:** Implementação de um modelo personalizado baseado em padrões individuais dos usuários, sem a necessidade de dados pré-rotulados;
- **Análise contextualizada:** Consideração do contexto circunstancial como fator determinante na interpretação dos sinais de estresse;
- **Aplicabilidade real:** Projeto especialmente adaptado para cenários com conectividade intermitente e variabilidade interindividual nos padrões de estresse.

Tais aspectos tornam a proposta deste trabalho mais adequada para aplicações no mundo real, onde a conectividade pode ser intermitente e os padrões de estresse variam significativamente entre indivíduos.

### 3. Materiais e Métodos

Para garantir a máxima eficiência e precisão da arquitetura proposta é essencial adotar uma abordagem multifacetada, abordando diferentes áreas de desenvolvimento. A complexidade envolvida na aplicação de modelos de *machine learning* em contextos reais exige não apenas a otimização do modelo em si, mas também melhorias na coleta e no pré-processamento dos dados, no desempenho das ferramentas de visualização e na escalabilidade da solução.

A arquitetura (Figura 1) segue o paradigma Edge-Fog-Cloud, que permite um processamento distribuído dos dados coletados pelos dispositivos vestíveis, otimizando o tempo de resposta e a eficiência do sistema. Esse paradigma é ideal para aplicações que lidam com dados de sensores em tempo real, como os batimentos cardíacos, pois evita a necessidade de enviar todas as informações para a nuvem constantemente, reduzindo latência, economizando largura de banda e garantindo respostas rápidas ao usuário.



**Figura 1. Representação da arquitetura proposta usando o paradigma Edge-Fog-Cloud**

### **3.1. Camada Edge**

A primeira etapa da arquitetura ocorre na camada Edge, que compreende os dispositivos mais próximos do usuário, como as smartbands conectadas aos smartphones. Esses dispositivos são responsáveis por coletar os sinais vitais do usuário em tempo real, registrando batimentos cardíacos com alta frequência, além de capturar informações contextuais como a localização geográfica (latitude e longitude) e o horário exato da medição, sendo todas essas informações importantes para o treinamento do modelo.

A smartband se conecta ao smartphone via *Bluetooth Low Energy* (BLE), garantindo uma comunicação contínua sem necessidade de conexão com a internet. Essa abordagem é eficiente, pois permite a coleta independente e descentralizada dos dados.

### **3.2. Camada Fog**

A camada Fog utiliza os conceitos da arquitetura Lambda em sua composição. Por isso, nesta camada, o dado segue dois fluxos distintos, um para o processamento interno e outro para o envio do dado para a nuvem. Assim, a camada Fog tem duas funções: (1) armazenamento local temporário e submissão do dado para predição de momentos de estresse, e (2) coleta e encaminhamento dos dados ao processamento na nuvem.

O modelo preditivo treinado fica armazenado na nuvem e na Fog no smartphone do usuário. Assim, quando um dado chega na camada Fog, ele é temporariamente armazenado, pré-processado e apresentado ao modelo preditivo, previamente treinado na nuvem, para realizar inferências em tempo real. O modelo interpreta esse dado e o classifica como sendo um momento de estresse ou não. O mesmo dado utilizado para predição de estresse na camada Fog é encaminhado para a nuvem.

Diferente de abordagens que realizam treinamento dos modelos preditivos localmente, nesta proposta o smartphone na Fog transmite os dados para a nuvem (linhas pontilhadas). O smartphone atua como um intermediário entre a smartband e a nuvem, garantindo que os dados coletados sejam enviados de maneira contínua e segura. Como os dispositivos móveis possuem restrições de energia e processamento, a decisão de não realizar treinamento na Fog otimiza o desempenho e prolonga a autonomia da bateria.

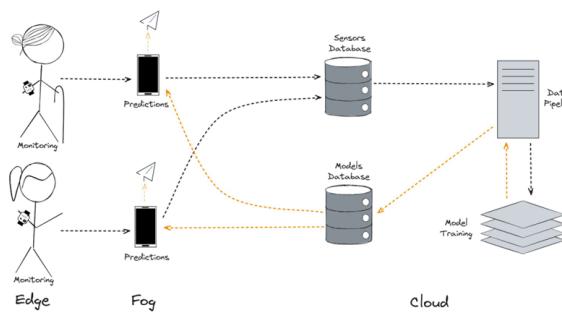
Caso o dispositivo esteja temporariamente sem conexão, os dados ficam armazenados localmente. Assim que a conectividade é restabelecida os dados são enviados para a nuvem e removidos do banco de dados local na Fog. Esse mecanismo garante que nenhuma informação seja perdida e que o monitoramento do estado físico e emocional do usuário continue sem interrupções.

### **3.3. Camada de Nuvem**

O processamento dos dados e treinamento dos modelos preditivos ocorrem de forma contínua na nuvem. Os dados enviados pela Fog são armazenados em um DataLake na nuvem. Posteriormente, os dados são consumidos do DataLake e é realizado o pré-processamento, transformando as medições brutas em um formato adequado para a etapa de treinamento. Esse processo inclui a normalização dos valores, a remoção de ruídos e a fusão de dados provenientes de diferentes medições, tornando a informação mais coesa e representativa do estado do usuário. Técnicas como *StandardScaler*, o uso da biblioteca de *deep learning* “TensorFlow” e da biblioteca “Pandas” foram cruciais para essa etapa. A representação global do funcionamento do projeto pode ser visualizada na Figura 2.

Com base nos trabalhos da literatura, inicialmente foi utilizado o modelo K-Neighrest Neighbors (KNN). Porém, considerando a natureza dos dados e resultados preliminares [Sergio et al. 2023], neste trabalho foi feito o uso de *Long Short Term Memory* (LSTM), um modelo baseado em série temporais. Acreditamos que o uso de LSTM seja mais adequado para a resolução do problema, visto que o batimento cardíaco no tempo  $t$  está relacionado com os batimentos em  $t - 1, t - 2, \dots$  e com a localização geográfica.

Os modelos treinados para cada usuário são armazenados na nuvem e enviados para a Fog. Assim, a nuvem armazena todos os modelos e a Fog armazena apenas o último modelo treinado do usuário. Essa abordagem melhora a precisão das previsões, pois os modelos podem ser continuamente aprimorados com novos dados e ajustados com maior poder computacional na nuvem.



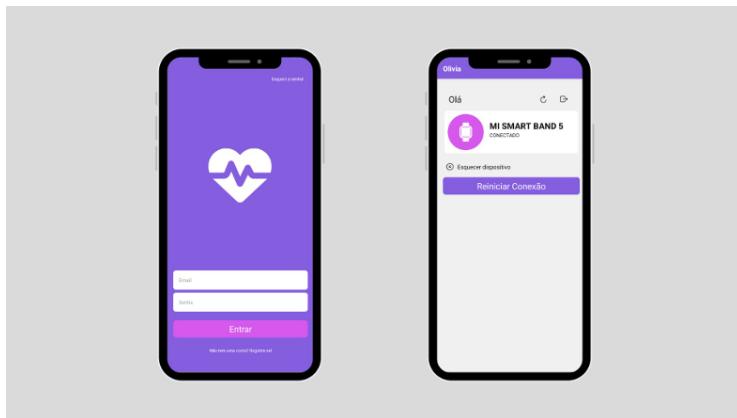
**Figura 2. Representação do funcionamento do aplicativo**

Ao dividir o foco da arquitetura em camadas estratégicas baseadas no paradigma Edge-Fog-Cloud, é possível aprimorar cada etapa do processo, garantindo que os dados sejam tratados de maneira adequada, que os modelos sejam treinados com alta performance e que os resultados sejam acessíveis e comprehensíveis para diferentes usuários. Dessa forma, essa abordagem integrada não apenas fortalece a confiabilidade das previsões, mas também permite uma solução mais robusta e adaptável.

#### 4. Resultados

Com base na arquitetura, foi desenvolvido um aplicativo, chamado Olivia, com a intenção de informar os usuários, baseado na coleta de seus dados de batimentos cardíacos em tempo real, o momento que eles estão passando por estresse. O aplicativo gerencia a autenticação dos usuários, se conecta à pulseira inteligente para coletar dados de frequência cardíaca e emite alertas caso um padrão incomum seja identificado. Para operar corretamente, a aplicação necessita de permissões como acesso à localização em segundo plano, autorização para execução contínua e permissão para uso do Bluetooth.

Foi desenvolvida uma tela de login onde os usuários inserem e-mail e senha para acessar o sistema. Após a autenticação, são redirecionados para a tela de monitoramento da conexão, onde podem emparelhar a pulseira para medições da frequência cardíaca (3). Nesta tela, os usuários selecionam “Conectar Dispositivo”, fazendo com que o aplicativo exiba uma lista de dispositivos próximos. Além disso, o aplicativo utiliza o GPS do smartphone para acompanhar mudanças de localização. Os dados são armazenados temporariamente no banco de dados local no dispositivo, que é um requisito da Fog.



**Figura 3. Aplicativo Olivia**

A sincronização com o banco de dados ocorre sob demanda, dependendo da disponibilidade da conexão. Foi utilizado o MongoDB no desenvolvimento do aplicativo. Caso a conexão com a pulseira inteligente seja perdida, o aplicativo executa uma tarefa em segundo plano para localizar e restabelecer a conexão com o dispositivo previamente registrado. Essa conexão de sincronização é realizada por meio do protocolo HTTPS para garantir a segurança dos dados. Os dados armazenados no banco de dados são marcados apenas com o ID para garantir a anonimidade.

Uma etapa importante é a notificação ao usuário quando ocorre o momento de estresse. Essas mensagens auxiliam o usuário a reconhecer eventos específicos que contribuem para o estresse pessoal no dia a dia. A notificação enviada pelo aplicativo Olivia ao usuário tem o seguinte formato: *Observei uma variação significativa em sua frequência cardíaca em [date] em [time]. Isso pode sugerir um momento de estresse. Recomendamos que você tire um tempo para relaxar e pratique técnicas de redução do estresse.*

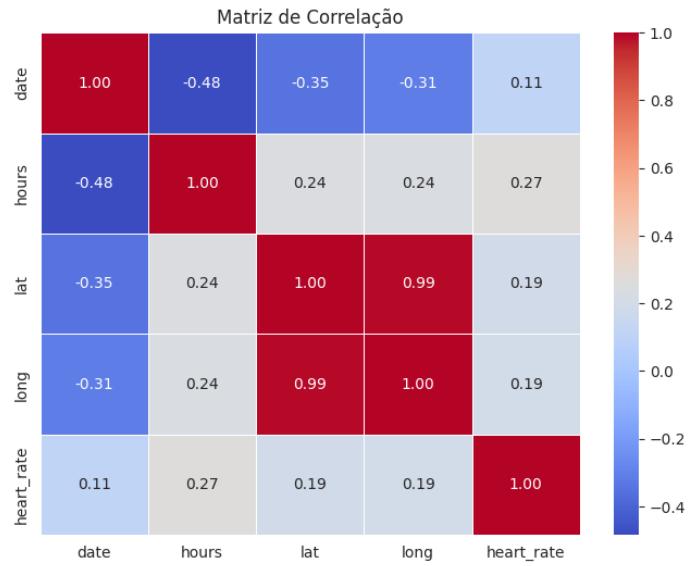
#### 4.1. Desenvolvimento do Modelo Preditivo

A conexão com a internet sendo estabelecida, os dados brutos são enviados para a nuvem e passam por um processo de normalização utilizando StandardScaler, garantindo que todas as variáveis fiquem em uma escala apropriada antes do treinamento. Esse processo de normalização é aplicado separadamente aos dados de treino e de validação para evitar vazamento de informações entre os conjuntos.

Algumas métricas e correlações são traçadas para entender a importância de cada uma das features de dados existentes no modelo. Foi modelada uma matriz de correlação para entender como as features se relacionam (Figura 4).

Analizando a matriz da Figura 4, observa-se que:

- Correlação entre latitude (lat) e longitude (long): Há uma correlação muito alta (0.99), o que indica que essas variáveis estão fortemente relacionadas, devido à proximidade geográfica dos pontos analisados.
- Correlação entre data e outras variáveis: A data tem correlações negativas com horas (-0.48), latitude (-0.35) e longitude (-0.31), o que confirma o padrão temporal nos dados, demonstrando que as coletas anteriores influenciam bastante no comportamento das subsequentes.



**Figura 4. Representação do funcionamento do aplicativo**

- Relação entre frequência cardíaca (heartrate) e outras variáveis:
  - Apresenta uma correlação positiva de 0.27 com as horas do dia, o que pode sugerir que a frequência cardíaca tende a aumentar em determinados períodos do dia.
  - Tem correlações relativamente baixas com latitude (0.19) e longitude (0.19), o que sugere que a localização pode ter um leve impacto na variação da frequência cardíaca.

Após a normalização, os dados são organizados em sequências temporais de 30 passos, de modo que cada entrada do modelo contenha informações dos últimos 30 registros. Esse formato de entrada permite que o modelo aprenda padrões ao longo do tempo e faça previsões mais precisas. O conjunto de dados é dividido em dados de treinamento e validação para que o modelo seja avaliado em dados não vistos durante o aprendizado.

O fluxo de treinamento do modelo foi implementado utilizando um sensor na primeira tarefa do Airflow [Foundation 2024]. Esse sensor tem a função de monitorar a fila do RabbitMQ para verificar se uma nova mensagem foi adicionada. A verificação ocorre a cada 10 minutos, e o fluxo só avança quando uma nova mensagem é recebida. Assim, a cada nova mensagem, um novo DAG run é iniciado, onde cada execução representa uma instância específica de um *Directed Acyclic Graph* (DAG). Esse mecanismo permite que múltiplos modelos sejam treinados simultaneamente, possibilitando a escalabilidade horizontal da aplicação com execuções paralelas de treinamento.

No fluxo de treinamento, após a recepção da mensagem, a tarefa de treinamento acessa os dados pré-processados do fluxo de processamento. A mensagem recebida na fila é um JSON serializado contendo informações sobre o intervalo de dados a ser utilizado no treinamento, bem como o identificador do usuário. Após a leitura desses dados, que estão armazenados em uma tabela no Delta Lake, eles são submetidos a um pré-processamento antes de serem inseridos no modelo. A entrada do modelo consiste em uma sequência de 30 registros contendo hora, longitude, latitude e frequência cardíaca, com o objetivo de prever a próxima frequência cardíaca da série.

Após o pré-processamento, os dados são utilizados para treinar o modelo. Durante essa etapa, um experimento é criado no MLFlow para registrar os parâmetros de configuração do modelo e as métricas extraídas do treinamento. O conjunto de dados é dividido em 71,5% para treinamento e 28,5% para validação, seguindo exemplos da documentação do Keras [Chollet 2015]. A arquitetura da rede consiste em duas camadas LSTM, cada uma com 64 unidades, seguidas por camadas de Dropout com taxa de 20% para reduzir o risco de overfitting. A camada final é uma camada Dense, responsável por gerar a previsão da frequência cardíaca. O modelo é compilado utilizando o otimizador RMSprop com taxa de aprendizado de 0.002 e é treinado minimizando o erro quadrático médio (*mean squared error*), garantindo estabilidade no processo de aprendizado. As informações sobre essa etapa estão consolidadas na tabela da Figura 5.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 30, 64)	17,664
dropout (Dropout)	(None, 30, 64)	0
lstm_1 (LSTM)	(None, 64)	33,024
dropout_1 (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65

Total params: 50,753 (198.25 KB)  
Trainable params: 50,753 (198.25 KB)  
Non-trainable params: 0 (0.00 B)

**Figura 5. Sumário do modelo**

Os valores de *loss* e *val loss* apresentados na Tabela 1 mostram a evolução do erro durante o treinamento do modelo. Observa-se uma redução consistente do erro de treinamento (*loss*), passando de 0.8170 na primeira época para valores abaixo de 0.6 a partir da sexta época, indicando um aprendizado eficaz. Além disso, a validação (*val loss*) apresenta uma tendência geral de redução, com destaque para a sétima época, onde atinge 1.0565, demonstrando a capacidade do modelo de capturar padrões relevantes nos dados. Esses resultados sugerem um bom ajuste do modelo ao problema proposto, com potencial para refinamento e otimização adicionais.

Época	Loss	Val Loss
1	0.8170	1.4513
2	0.6026	1.4021
3	0.6813	1.2196
4	0.5687	1.2445
5	0.6755	1.1280
6	0.5878	1.1259
7	0.6067	1.0565
8	0.5842	1.1695
9	0.6153	1.2037
10	0.5572	1.1694
11	0.6154	1.0956
12	0.5966	1.1469

**Tabela 1. Valores de loss e val loss por época durante o treinamento do modelo.**

Após o treinamento no experimento do MLFlow, as métricas são analisadas. Caso a taxa de erro do treinamento (MSE) seja inferior à taxa de erro de validação, o modelo não é registrado, e o usuário continua utilizando o último modelo validado. Se a MSE obtida for inferior a 70%, o modelo recém-treinado é registrado. Esse valor foi determinado empiricamente, analisando o comportamento da série e a adaptação do modelo aos dados, uma vez que o objetivo é identificar momentos que destoam dos padrões aprendidos.

#### **4.2. Discussões e Desafios**

Um dos maiores desafios é a melhoria do modelo de machine learning/IA, onde buscamos otimizar a performance dos algoritmos utilizados na análise e previsão de padrões fisiológicos, além de testar outros modelos e parâmetros da rede neural e do aprendizado profundo.

Outra abordagem que deve ser considerada é o desenvolvimento de mais funcionalidades no ambiente mobile. Criação de mais features que possam auxiliar mais os usuários a entender o que pode estar influenciando os seus respectivos momentos de estresse seria uma excelente contribuição.

Além disso, existem possibilidades de melhorias na captação de mais informação da pulseira e o tratamento desses novos dados coletados. Pressão arterial, oxigenação no sangue, temperatura corporal, dados de atividades físicas (por exemplo passos diários, distância percorrida, calorias queimadas), duração do sono e outros são algumas das possíveis features capaz de serem adicionadas ao modelo, aumentando com a precisão e aproximando mais de cenários reais.

### **5. Considerações Finais e Trabalhos Futuros**

Este trabalho propõe uma arquitetura para o monitoramento de usuários, com o propósito de detectar momentos de estresse. Para tal, foram usados dados de dispositivos IoT e aprendizagem profunda. O desenvolvimento do aplicativo Olivia, baseando-se na arquitetura, produziu resultados que mostram que é possível detectar momentos de estresse.

Além dos desafios destacados, como próximos passos, buscamos implementar um treinamento federado, uma abordagem inovadora que descentraliza a coleta e o processamento dos dados. Em vez de armazenar todas as informações em um servidor central, cada dispositivo ou "cliente" realiza o treinamento localmente, preservando a privacidade dos dados e reduzindo a necessidade de grandes transferências de informação.

Além disso, essa abordagem possibilita a personalização dos modelos, garantindo que cada sistema treinado localmente contribua para um aprendizado global mais eficiente e representativo. Com a descentralização, conseguimos capturar variações individuais de forma mais natural e adaptável, resultando em previsões mais precisas e um processo de treinamento mais escalável e sustentável.

Embora o aplicativo Olivia tenha sido funcionalmente testado e validado quanto à sua capacidade de capturar dados de localização e batimentos cardíacos para prever situações de possível estresse, ainda se faz necessária uma avaliação mais ampla da experiência do usuário. Testes realizados com os próprios autores indicaram boa precisão na geração de alertas, mas não foram suficientes para inferir que o usuário está realmente passando por estresses em todos esses momentos anormais. Estudos futuros devem considerar a realização de testes controlados com usuários reais, visando coletar feedback

qualitativo e quantitativo sobre aspectos como confiabilidade dos alertas, impacto na rotina e taxa de adesão ao uso do aplicativo.

Outro aspecto importante a ser explorado futuramente é a análise detalhada do impacto do Olivia no desempenho energético e computacional dos dispositivos envolvidos. A arquitetura Edge-Fog-Cloud traz vantagens em termos de escalabilidade, mas também pode introduzir desafios relacionados ao consumo de bateria do smartphone. É necessário avaliar, por exemplo, como o intervalo de amostragem dos batimentos cardíacos e a frequência de envio de dados afetam a duração da bateria do celular, além de examinar o uso de CPU e memória em cada camada da arquitetura. Investigações nessa direção permitirão otimizar o balanceamento entre precisão dos modelos preditivos e eficiência energética, fortalecendo a aplicabilidade da solução no contexto computacional.

## Agradecimentos

Agradecemos o apoio da Universidade Federal de Juiz de Fora no desenvolvimento deste trabalho. O autor Victor Ströele agradece o apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio em bolsa de produtividade em pesquisa 313568/2023-5 (PQ-2).

## Referências

- Bauer, M. E. (2002). Estresse. *Ciência hoje*, 30(179):20–25.
- Can, Y. S., Chalabianloo, N., Ekiz, D., Fernandez-Alvarez, J., Repetto, C., Riva, G., Iles-Smith, H., and Ersoy, C. (2020). Real-life stress level monitoring using smart bands in the light of contextual information. *IEEE Sensors Journal*, 20(15):8721–8730.
- Chollet, F. (2015). Keras. Accessed: 2024-05-15.
- Dieleman, J. L., Cao, J., Chapin, A., Chen, C., Li, Z., Liu, A., Horst, C., Kaldjian, A., Matyasz, T., Scott, K. W., et al. (2020). Us health care spending by payer and health condition, 1996-2016. *Jama*, 323(9):863–884.
- Foundation, A. S. (2024). Apache airflow.
- Gedam, S. and Paul, S. (2021). A review on mental stress detection using wearable sensors and machine learning techniques. *IEEE Access*, 9:84045–84066.
- Nurmi, J. and Lohan, E. S. (2021). Systematic review on machine-learning algorithms used in wearable-based ehealth data analysis. *IEEE Access*, 9:112221–112235.
- Risch, M., Grossmann, K., Aeschbacher, S., Weideli, O. C., Kovac, M., Pereira, F., Wohlwend, N., Risch, C., Hillmann, D., Lung, T., et al. (2022). Investigation of the use of a sensor bracelet for the presymptomatic detection of changes in physiological parameters related to covid-19: an interim analysis of a prospective cohort study (covi-gapp). *BMJ open*, 12(6):e058274.
- Sergio, W. L., di Iorio Silva, G., Ströele, V., and Dantas, M. A. R. (2023). *An Architecture Proposal to Support E-Healthcare Notifications*, page 157–170. Springer International Publishing.
- Zhao, L., Niu, X., Wang, L., Niu, J., Zhu, X., and Dai, Z. (2023). Stress detection via multimodal multitemporal-scale fusion: A hybrid of deep learning and handcrafted feature approach. *IEEE Sensors Journal*, 23(22):27817–27827.