MINISTÉRIO DA EDUCAÇÃO
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Engenharia de Produção

Análise comparativa de modelos de machine learning na identificação de anomalias comportamentais em vacas utilizando dados simulados e reais

Italo da Rocha Moreira
Marcelo Gustavo Sanchez Hidalgo

João Monlevade, MG
2025

Italo da Rocha Moreira
Marcelo Gustavo Sanchez Hidalgo

Análise comparativa de modelos de machine learning na identificação de anomalias comportamentais em vacas utilizando dados simulados e reais

Trabalho de conclusão de curso apresentado ao curso de Engenharia de Produção do Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto, como parte dos requisitos necessários para a obtenção do título de Engenheiro de Produção.

Orientador: Prof. Dr. Thiago Augusto De Oliveira Silva

João Monlevade, MG

2025

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

H632a Hidalgo, Marcelo Gustavo Sanchez.
Análise comparativa de modelos de machine learning na identificação de anomalias comportamentais em vacas utilizando dados simulados e reais. [manuscrito] / Marcelo Gustavo Sanchez Hidalgo. Italo da Rocha Moreira. - 2025.
28 f.: il.: color., gráf., tab.. + Algoritmo em pseudocódigo.

Orientador: Prof. Dr. Thiago Augusto de Oliveira Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Produção .

1. Aprendizado do computador. 2. Modelos matemáticos. 3. Probabilidades. 4. Simulação (Computadores). 5. Vacas - anomalias - comportamento. I. Moreira, Italo da Rocha. II. Silva, Thiago Augusto de Oliveira. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 519.8

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE ENGENHARIA DE PRODUCAO - ICEA

FOLHA DE APROVAÇÃO

Italo da Rocha Moreira

Marcelo Gustavo Sanchez Hidalgo

Análise comparativa de modelos de machine learning na identificação de anomalias comportamentais em vacas utilizando dados simulados e reais

Monografia apresentada ao Curso de Graduação em Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção

Aprovada em 10 de abril de 2025

Membros da banca

Prof. Dr. Thiago Augusto de Oliveira Silva - Orientador(a) - Universidade Federal de Ouro Preto

Prof. Dr. Alexandre Xavier Martins - Universidade Federal de Ouro Preto

Prof. Dr. Samuel Martins Drei - Universidade Federal de Ouro Preto

Thiago Augusto de Oliveira Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 10/04/2024

Documento assinado eletronicamente por **Thiago Augusto de Oliveira Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 10/04/2025, às 20:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0894401** e o código CRC **1B0782CB**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0817 - www.ufop.br

Agradecimentos

Antes de tudo, expressamos nossa gratidão às nossas famílias, cujo apoio incondicional foi essencial ao longo desta jornada.

Agradecemos imensamente a todas as pessoas que contribuíram de alguma forma ou outra para a realização deste objetivo, em especial à República Kalango e à República Cativeiro, que tiveram um papel fundamental na nossa formação pessoal. Somos gratos pelas amizades cultivadas e pelo suporte contínuo em cada etapa do caminho.

Também reconhecemos a importância da Universidade Federal de Ouro Preto (UFOP) e de seus professores, em especial Alexandre Xavier Martins e Paganini Barcellos de Oliveira, que compartilharam conhecimento e nos apoiaram em nossos projetos acadêmicos. Manifestamos ainda nossa gratidão ao professor orientador Thiago Augusto de Oliveira Silva, cuja dedicação e paciência foram fundamentais durante todo o processo.

Resumo

Este trabalho de conclusão de curso explora a aplicação de um simulador para gerar dados sobre o comportamento de vacas leiteiras, com o objetivo de prever a probabilidade de uma vaca estar ou não doente. O simulador já foi desenvolvido em um trabalho anterior e será utilizado para criar uma base de dados abrangente que inclui tanto vacas saudáveis quanto vacas doentes. A metodologia envolve a aplicação de modelos de *machine learning*, tais como Regressão Logística, *Random Forest*, *LSTM* e *SVM* para classificar as vacas em duas categorias: "saudável"e "doente", com base em variáveis comportamentais como alimentação, descanso e movimentação. A simulação é realizada a partir de dados reais coletados em uma fazenda na França, os quais são processados para ajustar o modelo preditivo. Os modelos serão testados utilizando tanto os dados simulados quanto os dados reais, em ambos os casos, esses dados serão usados para treinar e testar o modelo. A eficácia do modelo será avaliada por meio de métricas de desempenho como acurácia e precisão. O uso do simulador permite expandir a base de dados, podendo assim testar vários tamanhos de dados para avaliar qual o melhor para a identificação do estado de saúde de uma vaca.

Palavras-chave: Simulação; *Machine learning*; Modelagem.

Resumen

Esta tesis explora la aplicación de un simulador para generar datos sobre el comportamiento de vacas lecheras, con el objetivo de prever la probabilidad de que una vaca esté o no enferma. El simulador ya fue desarrollado en un trabajo anterior y será utilizado para crear una base de datos amplia que incluya tanto vacas saludables como vacas enfermas. La metodología envuelve la aplicación de modelos de *machine learning*, tales como Regresión Logística, *Random Forest*, *LSTM* y *SVM*, para clasificar las vacas en dos categorías: "saludable" y "enferma", con base en variables comportamentales como alimentación, descanso y movimiento. La simulación es realizada a partir de datos reales colectados en una hacienda en Francia, los cuales son procesados para ajustar el modelo predictivo. Los modelos serán probados utilizando tanto los datos simulados como los dados reales; en ambos casos, esos datos serán usados para entrenar y probar el modelo. La eficacia del modelo será medida por medio de métricas de desempeño como acuracia y precisión. El uso del simulador permite expandir la base de datos, pudiendo así probar varios tamaños de datos para evaluar cuál es el mejor para la identificación del estado de salud de una vaca.

Palabras clave: Simulación; *Machine learning*; Modelación.

Lista de ilustrações

Lista de tabelas

Lista de algoritmos

Sumário

1 Introdução

À medida que a tecnologia avança, novas possibilidades emergem, e a simulação é uma área de destaque. Este Trabalho de Conclusão de Curso dá continuidade à pesquisa iniciada por Moreira (2023), que se concentrou na criação de um simulador para monitorar as atividades diárias de vacas. O simulador já desenvolvido será utilizado para gerar um grande volume de dados sobre vacas doentes e saudáveis, uma vez que a quantidade de dados pode influenciar na acurácia de um modelo de predição.

Todos os animais têm naturalmente atividades conhecidas como essenciais. No caso de uma vaca leiteira, há atividades pontuais, como a ordenha, por exemplo (Wagner, 2020). Considerando apenas as atividades que não têm influência humana, foram levadas em conta atividades como comer, descansar e se mover.

Nesta ótica, assim como nos seres humanos, geralmente há uma duração média diária dedicada a atividades como comer. Por exemplo, no caso de um ser humano, se ele normalmente dorme 8 horas por dia, mas dormiu apenas 5 horas em certos dias, isso pode ser devido a vários fatores, como estresse no trabalho, uma doença, etc.

É necessário frisar que existe uma relação entre o comportamento de uma vaca e seu estado de saúde. Pois isso, é conveniente analisar os dados comportamentais com o intuito de retirar alguma informação relevante daquilo, como neste caso, ver se a vaca está ou não saudável.

O comportamento da vaca tem uma relação direta com o estado de saúde dela (Tullo *et al.*, 2016). Identificar anomalias no comportamento torna-se essencial para poder agir preventivamente em caso de uma vaca apresentar um deterioro na sua saúde, e com isso, evitar maiores perdas na produção e melhorar a qualidade de vida do gado, implementando de medidas corretivas. Com um volume pré definido de dados, é possível realizar análises detalhadas e identificar padrões de comportamento típicos de vacas saudáveis, além de detectar anomalias ao comparar esses dados com os de vacas doentes.

Com relação os dados comportamentais, eles nem sempre vão estar aptos para serem utilizados diretamente, a questão é saber se esses dados anormais estão fora do padrão e se precisam de tratamento. Como será apresentado pelo trabalho na Seção 3.

1.1 Objetivo geral

O trabalho desenvolvido apresenta como objetivo principal o desenvolvimento e avaliação de modelos de *machine learning* capazes de prever a probabilidade de uma vaca estar doente com base em dados de comportamento mediante a avaliação e classificação dos dados inseridos no sistema. A avaliação é conduzida utilizando métricas como acurácia e precisão, comparando os resultados obtidos com dados simulados e reais.

1.1.1 Objetivos específicos

Para cumprimento do objetivo geral é necessário atender aos seguintes objetivos específicos:

- Adequar o simulador de Moreira (2023) para ser alimentado a partir de dados reais de vacas doentes e poder assim simular dados de vacas que apresentam anomalias;
- Construir modelos de *regressão logística, SVM (support vector machine), LSTM (long short-term memory), random forest*;
- Adaptar os dados obtidos do simulador e dados reais para poder alimentar os modelos propostos e alcançar assim o objetivo geral.

1.2 Contribuições

As contribuições do trabalho englobam o uso de simulação para recriar o comportamento de uma vaca, a avaliação do comportamento de uma vaca a partir de dados simulado e dados reais para a identificação de problemas no estado de saúde da mesma e o uso de quatro modelos preditivos binários previamente mencionados para conseguir tal objetivo.

1.3 Organização do Trabalho

Este trabalho está estruturado em cinco seções além desta introdução. A Seção 2 oferece uma revisão da literatura relevante, fornecendo a base teórica para o estudo. A Seção 3 descreve a metodologia empregada, incluindo a descrição e tratamento dos dados. A Seção 4 aborda os resultados encontrados e a avaliação deles, por meio de uma avaliação comparativa. Por fim, a Seção 6 discute as considerações finais do trabalho proposto.

2 Revisão da Literatura

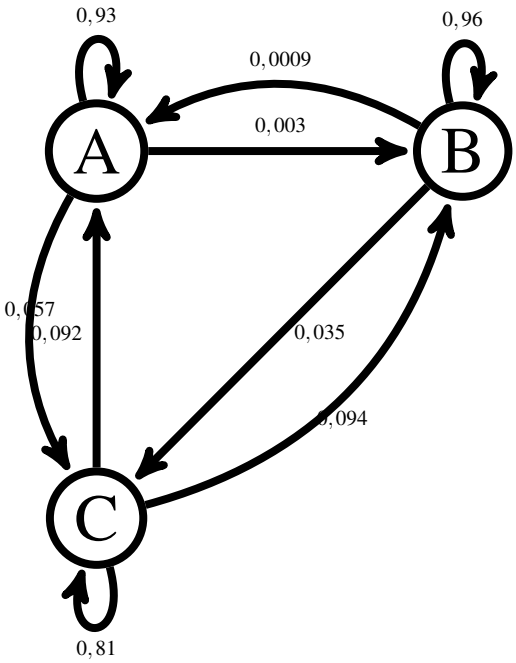
2.1 Cadeia de Markov

Uma Cadeia de Markov é um tipo de processo estocástico que descreve uma sequência de eventos em que a probabilidade de cada evento depende unicamente do evento imediatamente anterior (Souza, 2011).

Considerando as séries temporais diárias de comportamento das vacas, aplicou-se um processo de estado discreto utilizando Cadeias de Markov, uma técnica amplamente usada para análise de dados comportamentais (Araújo *et al.*, 2012). Estimou-se a probabilidade de transição entre os estados das atividades das vacas, como descanso, alimentação e movimento, ao longo do tempo.

Uma Cadeia de Markov (Figura 1) é caracterizada por um conjunto de estados e uma matriz de transição de probabilidades. Cada estado representa uma situação ou uma condição possível, e a matriz de transição indica as probabilidades de transição entre os estados (Rothe; Lames, 2022). O exemplo em questão, foi construído utilizando os dados reais da fazenda estudada. Neste caso, trata-se de uma Cadeia de Markov com probabilidades de transição válidas para todo um dia. Os estados A, B e C podem ser interpretados como *logette*: "estábulo individual ou baia", *auge*: "cocho"e *allée*: "corredor", respectivamente.

Figura 1 – Exemplo de Cadeia de markov com 3 estados.



As Cadeias de Markov geradas criaram varias matrizes de transição, com as probabilidades de se manter ou mudar de estado. Essas probabilidades foram utilizadas para a geração do simulador. Com o simulador gerado, ele cumpre seu papel nos fornecendo uma base de dados maior, que sera utilizada para o desenvolvimento deste trabalho de conclusão de curso (Guien *et al.*, 2023).

2.2 Simulação

Simulação é o processo de criação de um modelo abstrato que representa o comportamento de um sistema real, permitindo sua análise e experimentação sem a necessidade de manipular o sistema real diretamente (Prado, 2022). Ela envolve a imitação das operações e processos do sistema ao longo do tempo, utilizando dados e variáveis relevantes, com o objetivo de prever resultados, testar hipóteses, ou otimizar processos em ambientes controlados (SILVA, 2017). Simulações são amplamente utilizadas em diversas áreas, como ciências, engenharia, economia, e, neste caso, no monitoramento de comportamentos animais, como vacas em uma fazenda.

O processo de simulação é iniciado com a configuração das variáveis de tempo, como o tempo inicial, o número total de dias simulados, e outros parâmetros essenciais. Para garantir a reprodutibilidade dos resultados, o gerador de números aleatórios é inicializado com uma semente fixa.

A simulação é executada em ciclos diários. Em cada dia, o estado inicial da vaca é selecionado com base nas frequências das atividades fornecidas. Em seguida, para cada intervalo de tempo do dia, o estado da vaca é atualizado conforme as probabilidades de transição extraídas da matriz de transição. Essa abordagem permite simular a evolução dos estados ao longo do tempo.

A implementação foi realizada na linguagem de programação Python, utilizando as bibliotecas ‘pandas‘ e ‘numpy‘ para manipulação e análise de dados.

2.3 Modelo de Regressão Logística

A Regressão Logística consiste, de modo geral, em um modelo linear generalizado entre uma variável de resposta e uma ou mais variáveis explicativas (variáveis independentes) com base em um conjunto de dados categorizados, ou seja, amostras com um valor conhecido da variável de resposta (Zhou *et al.*, 2022). Esses dados categorizados são usados para ajustar um modelo de regressão, que pode ser utilizado para realizar predições.

Os modelos de Regressão Logística, basicamente, podem ser classificados de duas formas, segundo a variável de resposta: (i) Regressão Logística Binária (RLB), quando a saída é determinada por duas classes; (ii) Regressão Logística Multinomial (RLM), quando mais de duas classes são determinadas na saída (Campos *et al.*, 2022). Para este trabalho, será utilizado a RLB, onde as classes de saída são, de acordo com o objetivo do trabalho: "saudável"ou "doente". Sua expressão é dada pela Equação (2.1):

$$\log\left(\frac{p}{1-p}\right) = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n \tag{2.1}$$

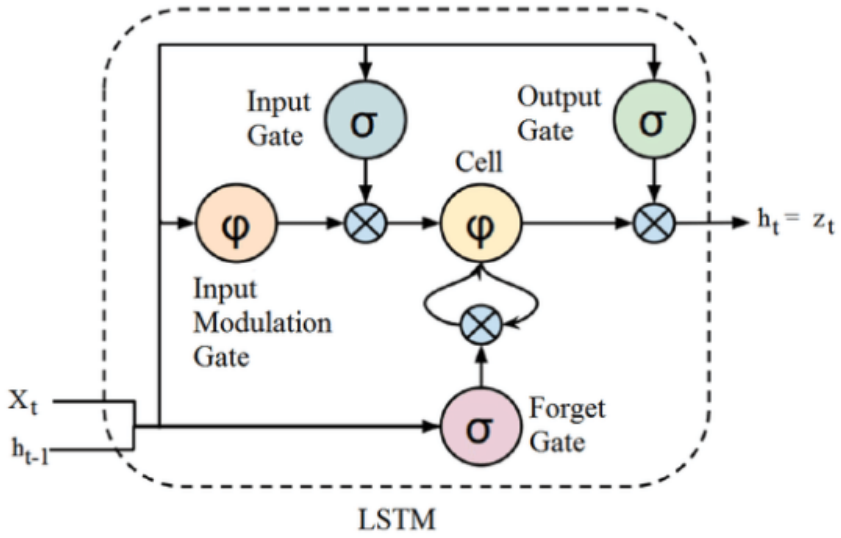
onde p é a probabilidade de sucesso, e os a_i são os coeficientes da regressão.

Como, por definição, a variável de resposta da RLB é dicotômica, é necessário determinar um limiar para a probabilidade que indicará a classe de saída. Este valor é conhecido como ponto de corte (James *et al.*, 2013), e é responsável por indicar se uma determinada amostra é classificada como "saudável"ou "doente".

2.4 Modelo *LSTM*

O modelo *LSTM* (Long Short-Term Memory) é uma variante derivada da Rede Neural Recorrente (RNN, pelas suas siglas em inglês), caracterizada por aumentar significativamente a capacidade de capturar dependências de longo prazo e reter memória vital em formação dentro de dados de séries temporais (Luo; Zhu; Li, 2025; Yu *et al.*, 2024). Ele pode aprender a conectar intervalos de tempo de mais de 1000 passos, mesmo em casos ruidosos, sequências de entrada compactas, sem perda de capacidades de atraso de tempo curto. A estrutura central do *LSTM* é composta pela porta de esquecimento, que decide quais informações antigas serão removidas da célula; a porta de entrada, que determina a quantidade de nova informação que será armazenada na célula e porta de saída, que controla a quantidade de informação que será utilizada para calcular a saída da célula (Zheng; Luo, 2024). Na Figura 2 é possivel observar a estrutura de uma célula do *LSTM*.

Figura 2 – Diagrama estrutural do *LSTM*.



Fonte: Adaptado de Gurumurthy e Jayalakshmi (2021).

2.5 Modelo *SVM*

O *SVM* (Support Vector Machine) pode ser usado para separar linearmente duas ou mais classes quando as duas se sobrepõem. O classificador estima o hiperplano ótimo que separa as diferentes classes da melhor maneira possível, maximizando a distância, também chamada de margem, até o ponto mais próximo de qualquer classe, também chamado de vetores de suporte (Vidal *et al.*, 2023). O método inclui uma transformação dos dados de seu espaço de recursos original em um espaço multidimensional (geralmente de dimensão infinita) onde os dados podem ser efetivamente separados por um hiperplano. Uma vez neste espaço de recursos transformado, dois hiperplanos paralelos representam os limites das duas classes (rotuladas y = +1 e y = - 1) (Vos *et al.*, 2022). Na Equação (2.2) é possível ver os dois hiperplanos sendo representados.

$$\begin{cases} \mathbf{w} \cdot \mathbf{x} - b = +1 \\ \mathbf{w} \cdot \mathbf{x} - b = -1 \end{cases} \tag{2.2}$$

2.6 Modelo *Random Forest*

Abreviado como RF, é outro classificador comumente usado. A ideia principal do algoritmo é construir uma infinidade de árvores de decisão (daí o termo forest, em inglês, 'floresta') usando subconjuntos de dados e características de formação selecionados aleatoriamente durante a etapa de formação (Liang *et al.*, 2024). Cada nó de uma árvore representa uma decisão baseada num dado input (neste caso, o estudo da vaca) que tem um valor acima ou abaixo de um determinado limiar. Ao prever os resultados, cada árvore na floresta classifica independentemente os dados de entrada. O modelo então agrega todos os resultados de todas as árvores para produzir a produção final. Numa floresta Aleatória, os resultados são as classes previstas pela maioria das árvores (Lardy; Ruin; Veissier, 2023). A RF é conhecida pela sua capacidade de lidar eficazmente com hiperparâmetros, resistir ao overfitting e gerir o desequilíbrio de classes (Li *et al.*, 2025).

2.7 Comportamento e Saúde das Vacas Leiteiras

A saúde das vacas leiteiras está intrinsecamente ligada à qualidade dos seus produtos e por conseguinte, a rentabilidade do empreendimento. Por exemplo, fatores que afetam a saúde das vacas, como dieta, paridade, fase de lactação e estação do ano, podem influenciar significativamente a composição de ácidos graxos do leite de vaca (Davis *et al.*, 2023).

Em relação ao comportamento, as vacas passam uma parte significativa do seu tempo deitadas (Churakov *et al.*, 2021). Quando os animais estão doentes, podem aparecer uma série de modificações comportamentais que levam a um estado letárgico durante o qual o animal reduz a sua atividade, dorme mais e em momentos quando estaria normalmente acordado (Wagner *et al.*, 2021).

Foi evidenciado que o tempo gasto pelas vacas deitadas ou alimentadas desempenha um papel importante em termos de produção de leite (Mattachini; Riva; Provolo, 2011). Por exemplo, alguns estudos apontam que uma vaca com problemas de saúde pode ter um aumento na taxa de alimentação resultante de uma diminuição no tempo gasto comendo e bebendo sem uma diminuição na ingestão de alimentos e bebidas (Benaissa *et al.*, 2023; Silberberg *et al.*, 2024).

As alterações de comportamento são indicadores claros dos problemas de saúde e de bem-estar das vacas leiteiras e, por conseguinte, podem ser utilizadas como elementos de apoio a um sistema de alerta precoce (Tullo *et al.*, 2016). De acordo com Wagner *et al.* (2021), o ritmo de atividade de uma vaca varia de um dia para o outro, mas é, no entanto, possível prever séries de 24 horas de atividade e detectar variações significativas (anomalias) nas mesmas utilizando algoritmos de aprendizado de máquina.

Com base nesses pressupostos, é possível viabilizar um estudo que relacione o comportamento de uma vaca leiteira com a possibilidade dela apresentar anomalias que configurem como problemas de saúde. Diante disso, na próxima seção, são descritos os próximos passos da pesquisa para alcançar os objetivos traçados.

3 Metodologia

A metodologia deste trabalho de conclusão de curso envolve o uso de alguns métodos de *machine learning* para predizer a probabilidade de uma vaca estar doente com base em dados simulados e reais de uma fazenda. Inicialmente, dados sobre o comportamento das vacas, como alimentação, descanso e movimento, serão coletados para vacas saudáveis e doentes. Para expandir a base de dados e aumentar a robustez do modelo, será utilizado um simulador criado em um trabalho anterior, que permitirá gerar um conjunto de dados mais amplo e diversificado.

Algoritmos como *SVM (Support Vector Machine), LSTM (Long Short-Term Memory), Random Forest e Regressão Logística* serão aplicados para identificar as relações entre as variáveis comportamentais e a presença de doenças, ajustando o modelo com base em um conjunto de treinamento e validando-o com um conjunto de teste. Métricas de desempenho, como acurácia e precisão, irão avaliar a eficácia do modelo.

A utilização do Python neste trabalho é justificada pela sua versatilidade, simplicidade e poder de processamento, especialmente em tarefas de análise de dados, modelagem estatística e simulação. Python possui uma vasta gama de bibliotecas especializadas, como *NumPy* e *Pandas*, que facilitam o tratamento de grandes volumes de dados.

Outra razão para escolher *Python* é sua popularidade no campo acadêmico e científico, o que garante uma extensa documentação, exemplos práticos e suporte da comunidade. Além disso, a flexibilidade da linguagem permite integrar diferentes etapas do projeto, desde a leitura e manipulação de dados, até a implementação de algoritmos de simulação e o ajuste de modelos preditivos (Borges, 2014).

3.1 Descrição dos dados

Para a construção deste trabalho, a base de dados utilizada considerou um rebanho de algumas dezenas de vacas. Cada uma delas está equipada com um sensor CowView, fornecido pela GEA Farm Technologies, com dimensões de aproximadamente 5 cm e preso ao seu coleira. Essas coleiras permitem determinar a posição da vaca a cada segundo por triangulação. A Figura 3 ilustra o tipo de coleira utilizada.

Figura 3 – Exemplo de coleira CowView.

Fonte: GEA Group AG.

As antenas receptoras são responsáveis por receber os sinais enviados pelas coleiras e transmiti-los para a estação base. Essas antenas podem ser instaladas em diferentes locais estratégicos da propriedade para garantir uma cobertura ampla, Figura 4.

A posição das vacas permite deduzir a atividade que elas estão realizando. Por exemplo, associa-se o fato de estar perto do cocho à vaca que está comendo, ou ainda a presença da vaca em uma baia, indicando que ela está descansando. Assim, a posição da vaca é registrada a cada segundo, e a principal atividade que ela realizou durante aquele tempo em questão é anotada.

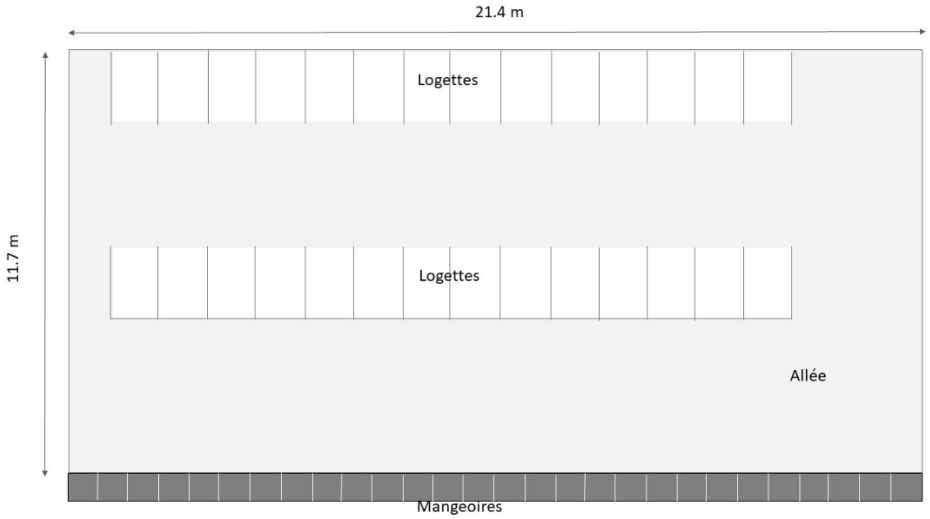
Dessa maneira, obtemos um conjunto de dados que indica, para cada vaca, a atividade principal que ela realizou a cada minuto, ao longo de um período de vários dias ou até mesmo de vários meses, aproximadamente 6 meses por vaca.

É importante notar que o modo de funcionamento da fazenda influencia o comportamento das vacas e, consequentemente, os dados coletados. É possível, por exemplo, reduzir artificialmente a iluminação durante a noite (Lardy *et al.*, 2022), distribuir a alimentação das vacas em horários fixos ou em intervalos, ou ainda implementar um sistema de ordenha automática das vacas que é limpo em horários fixos. É importante considerar que nem todas as fazendas utilizam os mesmos procedimentos, e a coleta de dados entre duas fazendas diferentes pode resultar em dados muito diferentes.

Figura 4 – Tipo de antena para determinar a posição e por consequência a atividade de uma vaca.

Para entender um pouco como esses sensores funcionam, é possível observar o plano do estábulo e uma foto real nas figuras 5 e 6 respectivamente. Esse conhecimento é importante, pois na tabela bruta de dados, existem 5 estados diferentes, mas apenas 3 setores de captura de atividades. Portanto, é necessário, seguindo o conselho de especialistas, agrupar as atividades para obter um melhor desempenho do simulador, e a forma como os dados foram tratados será abordada ao longo do trabalho. Na Figura 5, um *mangeoire* pode ser entendido como comedouro, *logettes* como baias ou e *allee* como corredores .

Figura 5 – Planta simplificada do estábulo.



Fonte: Adaptado de Wagner *et al.* (2020).

Figura 6 – Fotografia do estábulo estudado.

Finalmente, é importante ressaltar que, para o presente trabalho, é importante separar os dados entre vacas saudáveis e vacas com alguma anomalia, para após realizar o tratamento destes dados.

3.2 Tratamento de dados

O simulador utiliza duas principais fontes de dados: uma planilha contendo as frequências das atividades das vacas e outra com a matriz de transição das atividades. Além disso, são definidos parâmetros como o número de intervalos e o número mínimo de intervalos, que orientam a simulação.

Para a geração do simulador descrito no trabalho de Moreira (2023), o tratamento dos dados foi realizado a partir de uma planilha bruta, considerando inicialmente apenas uma única vaca devido à variação significativa nos hábitos individuais das vacas. Porém, pensando em aumentar a base de dados, os dados simulados foram construídos baseados nos dados coletados reais de todas as vacas em registro.

Durante a simulação, o estado atual da vaca (*logette, auge e allée*) é atualizado com base nas probabilidades de transição associadas a cada estado. O novo estado é então registrado para o período de tempo correspondente. Esse processo é repetido para todos os intervalos do dia e para cada dia de simulação.

Os estados simulados são armazenados em uma lista e, ao final da simulação, convertidos em um *DataFrame* para análise. O tempo total de execução da simulação é calculado e exibido juntamente com a tabela de resultados.

Ao usar uma planilha reduzida tratada (Valentin, 2022), foram observados resultados semelhantes. Esta planilha menor adota outros critérios para a limpeza dos dados, como a exclusão dos dias anteriores a uma situação anormal, como uma doença, com base na variação dos dias conforme a condição anômala apresentada pela vaca.

Para o simulador de vacas doentes, é essencial processar e alimentar o simulador com dados de vacas que apresentam doenças ou anomalias. Foram gerados dados a partir de períodos em que a vaca estava saudável, assim como dados equivalentes para vacas doentes. Na Figura 7, é possível ver uma imagem de uma vaca doente. Comportamentos anormais de vacas doentes podem impactar a qualidade do simulador. Portanto, dados inconclusivos e que não são doenças, como aqueles classificados como "Na"(não aplicável), cio, parto e distúrbios, foram excluídos.

Abaixo temos a Tabela 1 com os minutos passados em cada estado, considerando a base de dados sem tratamento.

Tabela 1 – Tempo em minutos gasto em cada estado

Estado	Tempo (minutos)
Cio	169920
Parto	36000
Claudicação	23040
Mastite	15840
LPS	77760
Acidose	0
Outras Doenças	34560
Acidentes	0
Distúrbio	240480
Mistura	0

As atividades foram agrupadas da seguinte forma:

- *ALLE*: A vaca está em pé, imóvel ou caminhando em um corredor.
- *LOGETTE*: A vaca está descansando em uma baia.
- *AUGE*: A vaca está comendo ou bebendo.

Com os dados filtrados, esta nova planilha foi utilizada para calibrar o simulador e alimentar os modelos de *machine learning*.

Figura 7 – Fotografia de uma vaca doente.

Fonte: Imagem retirada de Wagner (2020).

3.2.1 Pré-processamento dos dados

O pré-processamento de dados é uma etapa essencial para garantir a qualidade e a representatividade das informações utilizadas no estudo. Inicialmente, os dados das vacas saudáveis e doentes foram organizados em duas bases distintas: ‘dadosS‘ para vacas saudáveis e ‘dadosD‘ para vacas doentes. Em seguida, foi adicionada uma coluna denominada "dia", que representa o intervalo diário, obtido a partir do índice dividido por 1440 (número total de minutos em um dia). Essa etapa permitiu a segmentação dos dados com base na variação temporal, facilitando análises subsequentes.

Para a base de dados de vacas doentes, foram coletados 20 dias, divididos em 5 grupos de 4 dias consecutivos de vacas que apresentavam as anomalias apresentadas na Tabela 1, mais especificamente: mastite e LPS. Para equilibrar o conjunto de dados, foi necessário reduzir a quantidade de registros das vacas saudáveis, uma vez que o número de dias registrados para esse grupo era significativamente maior do que para as vacas doentes (213 dias de vacas saudáveis versus 20 dias de vacas doentes). Dessa forma, foram selecionados aleatoriamente 20 intervalos consecutivos de dias a partir da base de dados das vacas saudáveis, garantindo uma distribuição mais homogênea e minimizando possíveis vieses nos modelos preditivos. Essa seleção foi realizada de maneira aleatória, porém reprodutível, utilizando uma semente fixa para garantir consistência nos experimentos.

Após a seleção dos intervalos, foi realizada a transformação dos dados para análise em períodos de 30 minutos tanto para a base de dados de vacas saudáveis quamto para vacas doentes. A conversão da coluna *"hour"* para o formato datetime permitiu a segmentação das atividades em intervalos fixos dentro de cada dia. Em seguida, os dados foram agrupados por dia e intervalo de 30 minutos, contabilizando a ocorrência de cada atividade (*logette*, *auge* e *allee*). Posteriormente, esses valores foram convertidos em porcentagens, proporcionando uma visão relativa da distribuição de atividades ao longo do tempo, reduzindo assim o número de linhas a serem lidas pelos modelos, fazendo o processo mais eficiente e menos demorado. O formato final das bases de dados para treinar e testar os modelos tem a seguinte forma, mostrada na Tabela 2.

dia	intervalo	ALLEE	AUGE	LOGETTE
163	00:00:00	0.0	0.0	100.0
163	00:30:00	0.0	0.0	100.0
163	01:00:00	0.0	0.0	100.0
163	01:30:00	0.0	0.0	100.0
163	02:00:00	0.0	0.0	100.0
⋮	⋮	⋮	⋮	⋮
182	21:30:00	0.0	0.0	100.0
182	22:00:00	0.0	0.0	100.0
182	22:30:00	0.0	0.0	100.0
182	23:00:00	0.0	0.0	100.0
182	23:30:00	0.0	0.0	100.0

Tabela 2 – Estrutura padrão das bases de dados após tratamento

Por fim, os dados processados foram exportados para arquivos do formato *".xlsx"*, possibilitando a utilização em análises posteriores e na modelagem preditiva para identificação de padrões anômalos no comportamento das vacas. Os modelos preditivos que foram utilizados se alimentam com as duas bases de dados e realizam um tratamento de dados dentro deles, com o intuito estarem aptos para serem processados pelos modelos em questão. Tal tratamento de dados é apresentado no Algoritmo 1 e ele se aplica a todos os modelos aplicados.

Algoritmo 1: Tratamento de Dados

função *TratamentoDados(caminho_saudaveis, caminho_doentes)*
$dados\_saudaveis \leftarrow carregar\_dados(caminho\_saudaveis)$;
$dados\_doentes \leftarrow carregar\_dados(caminho\_doentes)$;
// Carrega os dados das vacas saudáveis e doentes a partir de um arquivo.
$dados\_saudaveis['target'] \leftarrow 0$;
$dados\_doentes['target'] \leftarrow 1$;
// Adiciona uma coluna 'target' com valor 0 para vacas saudáveis e 1 para doentes.
$dados \leftarrow combinar\_dados(dados\_saudaveis, dados\_doentes)$; **// Combina os dados em um único DataFrame.**
$dados \leftarrow remover\_nulos(dados)$; **// Remove linhas com valores nulos.**
$dados \leftarrow remover\_duplicatas(dados)$; **// Remove linhas duplicadas.**
$dados \leftarrow criar\_features(dados)$; **// Cria novas features, como hora e minuto, a partir de uma coluna de data/hora.**
$dados['hora'] \leftarrow dados['interval'].dt.hour$; **// Extrai a hora da coluna 'interval'.**
$dados['minuto'] \leftarrow dados['interval'].dt.minute$; **// Extrai o minuto da coluna 'interval'.**
$dados \leftarrow codificar\_categorias(dados)$; **// Codifica variáveis categóricas em valores numéricos.**
$dados \leftarrow normalizar(dados)$; **// Normaliza as features para que tenham a mesma escala.**
$(X\_train, X\_test, y\_train, y\_test) \leftarrow dividir\_dados(dados)$; **// Divide os dados em conjuntos de treino e teste.**
retorna $(X\_train, X\_test, y\_train, y\_test)$ **// Retorna os dados processados e divididos.**

O código apresentado implementa esse processo de pré-processamento. A função *"TratamentoDados"* recebe como entrada os caminhos dos arquivos contendo os dados das vacas saudáveis e doentes. Primeiramente, as informações são carregadas e armazenadas em variáveis distintas, *"dados_saudaveis"* e *"dados_doentes"*. Em seguida, adiciona-se a coluna *"target"*, permitindo que os modelos de aprendizado de máquina diferenciem as duas categorias de animais.

Após essa etapa, os dados são combinados em um único conjunto, garantindo que todas as informações estejam padronizadas para análises posteriores. A limpeza dos dados é realizada removendo-se valores nulos e duplicatas, assegurando que apenas informações relevantes sejam consideradas no modelo.

A criação de novas *features* ocorre com a extração de informações temporais, como hora e minuto, a partir da coluna de data/hora. Essa transformação auxilia na identificação de padrões comportamentais ao longo do dia. Além disso, as variáveis categóricas são convertidas em valores numéricos, possibilitando que os modelos estatísticos e de *machine learning* interpretem os dados corretamente.

A normalização das variáveis é outro passo crucial do processo, garantindo que todas as características possuam a mesma escala e evitando que diferenças de magnitude entre elas afetem os resultados. Finalmente, os dados são divididos em conjuntos de treino e teste, permitindo a avaliação do modelo antes de sua aplicação prática. O ajuste foi feito para incluir a proporção de 90% para treino e 10% para teste, a escolha está baseada no tamanho das bases a serem utilizadas, dando assim 18 dias para treinar os modelos e 2 dias para testar.

Com isso, o pré-processamento garante um conjunto de dados limpo, padronizado e pronto para ser utilizado na modelagem preditiva, sendo essencial para melhorar a qualidade da previsão dos modelos e obter resultados mais precisos na identificação de padrões anômalos no comportamento das vacas, além de reduzir o tamanho da base a ser lida e com isso, gerando um ganho em eficiência.

4 Resultados

A regressão logística foi pensada para este problema porque se trata de uma tarefa de classificação binária, ou seja, determinar se uma vaca está saudável ou doente com base em variáveis comportamentais. Esse modelo estatístico é capaz de estimar a probabilidade de um evento ocorrer a partir de dados de entrada, permitindo interpretar como cada variável influencia a predição do estado de saúde do animal. Além disso, a regressão logística é relativamente simples de implementar e interpretar.

A seguir, é apresentado a estrutura do método no Algoritmo 2:

Algoritmo 2: Regressão Logística

1 **função** *RegressaoLogistica(X_train, y_train, X_test, y_test)*
2 $modelo \leftarrow LogisticRegression(max\_iter = 1000)$; **// Cria um modelo de Regressão Logística com limite de 1000 iterações.**
3 $modelo.fit(X\_train, y\_train)$; **// Treina o modelo com os dados de treino.**
4 $y\_pred \leftarrow modelo.predict(X\_test)$; **// Faz previsões com os dados de teste.**
5 $acuracia \leftarrow accuracy\_score(y\_test, y\_pred)$; **// Calcula a acurácia do modelo.**
6 **retorna** *acuracia* **// Retorna a acurácia do modelo.**

O Algoritmo 2 apresenta a implementação da regressão logística para a classificação do estado de saúde das vacas. Inicialmente, um modelo de regressão logística é criado com um limite de 1000 iterações para garantir a convergência. Em seguida, o modelo é treinado utilizando os dados de treino (X_train, y_train) e, posteriormente, faz previsões sobre os dados de teste (X_test). A acurácia do modelo é então calculada comparando as previsões com os valores reais (y_test), e esse valor é retornado como medida de desempenho do modelo.

4.0.0.1 Modelo *LSTM*

O uso da rede neural *LSTM* é justificado neste problema porque os dados de comportamento das vacas possuem uma estrutura sequencial ao longo do tempo, onde o estado atual pode ser influenciado por estados anteriores. Diferente de modelos tradicionais como a regressão logística, que tratam cada observação de forma independente, a *LSTM* é capaz de capturar dependências temporais e padrões dinâmicos nos dados, permitindo uma análise mais precisa da evolução do comportamento das vacas. Isso é essencial para identificar mudanças sutis que antecedem a manifestação da doença, melhorando a capacidade preditiva do modelo.

A seguir, é apresentado a estrutura do método no Algoritmo 3:

Algoritmo 3: LSTM

função *LSTM(X_train, y_train, X_test, y_test)*
$seq\_length \leftarrow 48$; **// Define o comprimento das sequências temporais.**
$X\_train\_seq \leftarrow criar\_sequencias(X\_train, seq\_length)$; **// Transforma os dados de treino em sequências temporais.**
$X\_test\_seq \leftarrow criar\_sequencias(X\_test, seq\_length)$; **// Transforma os dados de teste em sequências temporais.**
$y\_train\_seq \leftarrow y\_train[seq\_length - 1:]$; **// Ajusta os rótulos de treino para o formato de sequências.**
$y\_test\_seq \leftarrow y\_test[seq\_length - 1:]$; **// Ajusta os rótulos de teste para o formato de sequências.**
$modelo \leftarrow Sequential()$; **// Cria um modelo sequencial.**
$modelo.add(LSTM(50, input\_shape = (X\_train\_seq.shape[1], X\_train\_seq.shape[2]))$; **// Adiciona uma camada LSTM com 50 unidades.**
$modelo.add(Dropout(0.2))$; **// Adiciona uma camada de Dropout para evitar overfitting.**
$modelo.add(Dense(1, activation =' sigmoid'))$; **// Adiciona uma camada de saída com ativação sigmoide.**
$modelo.compile(optimizer = Adam(learning\_rate = 0.001), loss =' binary\_crossentropy', metrics = ['accuracy'])$; **// Compila o modelo com otimizador Adam e função de perda binária.**
$modelo.fit(X\_train\_seq, y\_train\_seq, epochs = 5, batch\_size = 64, validation\_data = (X\_test\_seq, y\_test\_seq))$; **// Treina o modelo com os dados de treino.**
$y\_pred \leftarrow (modelo.predict(X\_test\_seq) > 0.5).astype("int32")$; **// Faz previsões com os dados de teste.**
$acuracia \leftarrow accuracy\_score(y\_test\_seq, y\_pred)$; **// Calcula a acurácia do modelo.**
retorna *acuracia* **// Retorna a acurácia do modelo.**

O algoritmo 3 implementa um modelo *LSTM* para classificar sequências temporais de dados. Primeiramente, os dados de entrada são transformados em sequências de comprimento 48 (número de linhas que representam os intervalo de 30 minutos ao longo do período de 1 dia) para capturar dependências temporais. Em seguida, um modelo sequencial é criado com uma camada *LSTM* de 50 unidades, seguida de uma camada *dropout* para evitar *overfitting*, que é um problema de aprendizado de máquina que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, fazendo com que o modelo não consiga fazer previsões precisas para novos dados.

Além disso, o modelo possui uma camada densa com ativação sigmoide para saída binária. O modelo é compilado com o otimizador Adam e a função de perda binária, sendo treinado por 5 épocas com *batch size (tamanho de lote)* de 64, sendo eles escolhidos após alguns testes com outros tamanhos de lote e outros números de épocas, tornando-se o mais eficiente, evitando assim ter que fazer manualmente uma parada antecipada *(early stopping)*. Após o treinamento, o modelo faz previsões nos dados de teste e calcula a acurácia, que é retornada como métrica de desempenho.

4.0.0.2 Modelo *SVM*

O uso do *SVM* para este problema se justifica pela sua capacidade de lidar com problemas de classificação, especialmente quando há uma separação não linear entre as classes, como pode ocorrer na distinção entre vacas saudáveis e doentes com base em padrões de comportamento. Além disso, o *SVM* é robusto em relação a dados de alta dimensionalidade, o que pode ser útil caso o conjunto de variáveis preditoras seja extenso. Com o uso de *kernels*, o *SVM* pode mapear os dados para um espaço de maior dimensão, permitindo encontrar um hiperplano ótimo de separação entre as classes. Por fim, essa técnica é conhecida por sua boa generalização, reduzindo o risco de *overfitting*, mesmo com um volume de dados limitado.

A seguir, é apresentado a estrutura do método no Algoritmo 4:

Algoritmo 4: SVM

função *SVM(X_train, y_train, X_test, y_test)*
$scaler \leftarrow StandardScaler()$; **// Normaliza os dados de treino e teste.**
$X\_train\_scaled \leftarrow scaler.fit\_transform(X\_train)$; **// Aplica a normalização nos dados de treino.**
$X\_test\_scaled \leftarrow scaler.transform(X\_test)$; **// Aplica a normalização nos dados de teste.**
$modelo \leftarrow SVC(kernel =' rbf', C = 1.0, gamma =' scale')$; **// Cria um modelo SVM com kernel RBF.**
$modelo.fit(X\_train\_scaled, y\_train)$; **// Treina o modelo com os dados de treino normalizados.**
$y\_pred \leftarrow modelo.predict(X\_test\_scaled)$; **// Faz previsões com os dados de teste normalizados.**
$acuracia \leftarrow accuracy\_score(y\_test, y\_pred)$; **// Calcula a acurácia do modelo.**
retorna *acuracia* **// Retorna a acurácia do modelo.**

O algoritmo 4 implementa um modelo *SVM* para classificação, iniciando com a normalização dos dados de treino e teste usando *StandardScaler* para melhorar o desempenho do modelo. Em seguida, um classificador *SVM* com *kernel RBF* é criado e treinado com os dados normalizados, permitindo que o SVM mapeie os dados de entrada para um espaço de características de maior dimensão, onde uma separação linear entre as classes pode ser mais facilmente encontrada. Após o treinamento, o modelo realiza previsões nos dados de teste e calcula a acurácia comparando as previsões com os rótulos reais, retornando esse valor como métrica de desempenho.

4.0.0.3 Modelo *Random Forest*

O uso do *Random Forest* para este problema se justifica pela sua capacidade de lidar com grandes volumes de dados e identificar padrões complexos em conjuntos de dados ruidosos, como os comportamentos das vacas. Como um modelo baseado em múltiplas árvores de decisão, ele reduz o risco de *overfitting*, garantindo maior generalização na predição do estado de saúde das vacas. Além disso, o *Random Forest* lida bem com variáveis categóricas e numéricas, sendo robusto para a análise de diferentes combinações de comportamentos, o que pode melhorar a precisão na detecção precoce de doenças.

A seguir, é apresentado a estrutura do método no Algoritmo 5:

Algoritmo 5: Random Forest

função *RandomForest(X_train, y_train, X_test, y_test)*
$modelo \leftarrow RandomForestClassifier(random\_state = 42)$; **// Cria um modelo de Random Forest com 100 árvores.**
$modelo.fit(X\_train, y\_train)$; **// Treina o modelo com os dados de treino.**
$y\_pred \leftarrow modelo.predict(X\_test)$; **// Faz previsões com os dados de teste.**
$acuracia \leftarrow accuracy\_score(y\_test, y\_pred)$; **// Calcula a acurácia do modelo.**
retorna *acuracia* **// Retorna a acurácia do modelo.**

O código implementa uma função chamada *RandomForest*, que utiliza o algoritmo de *Random Forest* para classificar dados. Primeiro, cria um modelo de *Random Forest* com 100 árvores, utilizando a classe *RandomForestClassifier* e definindo uma semente aleatória para garantir reprodutibilidade. Em seguida, o modelo é treinado com os dados de entrada *X_train* (características de treinamento) e *y_train* (rótulos de treinamento). Após o treinamento, o modelo faz previsões sobre o conjunto de teste *X_test* e as compara com os rótulos reais *y_test*. A acurácia do modelo é calculada com a função *accuracy_score*, que avalia a precisão das previsões, e essa acurácia é retornada como resultado.

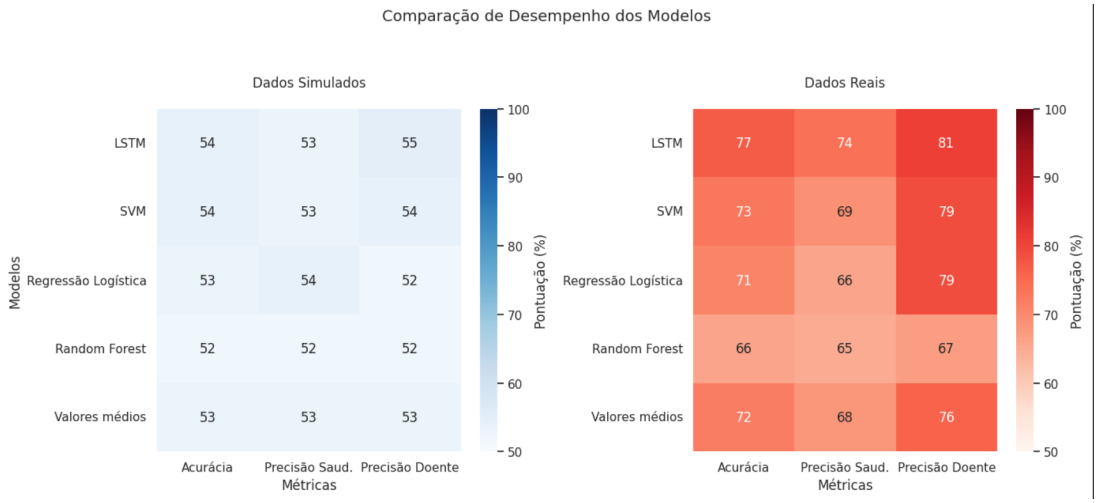
4.0.0.4 Resultados dos Modelos

Tabela 3 – Resultados dos Modelos utilizando dados simulados e reais

Modelo	Dados Simulados			Dados Reais		
	Acurácia (%)	Precisão (Saud.)	Precisão (Doente)	Acurácia (%)	Precisão (Saud.)	Precisão (Doente)
LSTM	**54,00**	0,53	0,55	**77,00**	0,74	0,81
SVM	**54,00**	0,53	0,54	**73,00**	0,69	0,79
Regressão Logística	**53,00**	0,54	0,52	**71,00**	0,66	0,79
Random Forest	**52,00**	0,52	0,52	**66,00**	0,65	0,67
Valores médios	**53,00**	0,53	0,53	**71,75**	0,68	0,76

Os resultados apresentados na tabela comparam o desempenho de quatro modelos de aprendizado de máquina apresentados anteriormente em dois conjuntos de dados: dados simulados e dados reais. A avaliação foi realizada com base em três métricas principais: Acurácia, Precisão para casos saudáveis e Precisão para casos doentes. Esses resultados permitem uma análise detalhada do comportamento dos modelos em diferentes cenários.

Figura 8 – Gráfico representando resultados.



No caso dos dados simulados, todos os modelos apresentaram desempenho semelhante, com acurácia variando entre 52% e 54%. Essa uniformidade sugere que os dados simulados podem ser menos complexos ou menos variados, o que resulta em uma performance comparável entre os modelos. Além disso, as precisões para casos saudáveis e doentes foram equilibradas, indicando que os modelos não apresentaram viés significativo para nenhuma das classes. Esse cenário pode ser útil para testes iniciais, mas não reflete necessariamente a complexidade de dados reais. Uma outra justificativa para o baixo desempenho dos modelos ao utilizar os dados simulados é o fato de o simulador não representar fielmente 100% dos dados reais, apresentando uma acurácia de aproximadamente 75%. Essa eficácia foi ainda mais reduzida ao se utilizar os dados de todas as vacas, tratando o comportamento de todas como um padrão único, o que pode não refletir as variações individuais presentes na realidade. Além disso, a não consideração de períodos atípicos e temporais causados por sazonalidade também contribuiu para a diminuição do desempenho dos modelos, uma vez que esses fatores podem influenciar significativamente os padrões comportamentais e de saúde dos animais.

Por outro lado, ao analisar os dados reais, observa-se uma melhora significativa no desempenho dos modelos, com acurácia variando de 66% a 77%. O modelo *LSTM* destacou-se como o mais eficaz, alcançando uma acurácia de 77% e uma precisão de 0,81 para casos doentes. Esse resultado sugere que o *LSTM*, por ser uma rede neural recorrente, é capaz de capturar padrões temporais e complexos presentes nos dados reais, o que o torna particularmente adequado para problemas que envolvem sequências ou dependências temporais. O *SVM* também apresentou um bom desempenho, com acurácia de 73% e precisão de 0,79 para casos doentes, embora tenha ficado ligeiramente abaixo do *LSTM*. Isso pode ser atribuído à dificuldade do *SVM* em lidar com dados não lineares ou de alta dimensionalidade.

A Regressão Logística, por sua vez, obteve uma acurácia de 71% nos dados reais, com precisão de 0,79 para casos doentes. Embora seja um modelo mais simples, seu desempenho foi competitivo, especialmente na identificação de casos doentes. Já o *Random Forest*, apesar de ser um modelo robusto e amplamente utilizado, apresentou o menor desempenho entre os modelos testados, com acurácia de 66% e precisão de 0,67 para casos doentes. Esse resultado pode indicar que o *Random Forest* não conseguiu capturar tão bem os padrões complexos dos dados reais, especialmente em comparação com o *LSTM*.

Ao analisar os valores médios, observa-se que, para os dados simulados, a acurácia média foi de 53%, com precisão média de 0,53 para ambos os casos (saudáveis e doentes). Já nos dados reais, a acurácia média subiu para 71,75%, com precisão média maior para casos doentes (0,76) do que para casos saudáveis (0,68). Essa diferença reforça a ideia de que os dados reais contêm mais informações e padrões que os modelos conseguem explorar, além de indicar que os modelos tendem a ser mais precisos na identificação de casos doentes. Essa característica pode ser crucial dependendo da aplicação, especialmente em cenários onde a detecção de casos doentes é prioritária.

Os resultados demonstram que o *LSTM* é o modelo mais eficaz para dados reais, especialmente na identificação de casos doentes, devido à sua capacidade de lidar com dependências temporais e padrões complexos. Modelos mais simples, como Regressão Logística e *Random Forest*, apresentam desempenho inferior, mas ainda podem ser úteis dependendo do contexto e da complexidade dos dados. A diferença de desempenho entre dados simulados e reais ressalta a importância de utilizar dados representativos para a avaliação de modelos, garantindo que os resultados sejam aplicáveis em cenários práticos.

5 Considerações Finais

Com base nos resultados obtidos, conclui-se que, embora os dados simulados tenham sido úteis para expandir a base de treinamento do modelo, os dados reais demonstraram maior eficácia na avaliação do estado de saúde das vacas leiteiras. Isso sugere que, mesmo em menor quantidade, os dados coletados diretamente no ambiente de criação refletem melhor os padrões comportamentais dos animais, permitindo uma predição mais confiável. Esse resultado reforça a importância de aprimorar a coleta de dados reais, garantindo que o modelo opere com informações mais próximas da realidade do rebanho.

Além disso, a qualidade da predição pode ser significativamente melhorada com um aprofundamento no estudo do comportamento das vacas e com a adoção de novas técnicas de coleta de dados. Atualmente, as informações utilizadas no modelo são captadas por triangulação e localização e, posteriormente, interpretadas como atividades específicas. No entanto, essa abordagem pode apresentar limitações, desde que os dados obtidos por triangulação não registram diretamente as ações executadas pelos animais, apenas a dedução do que a vaca está fazendo a partir da posição em que ela se encontra. A utilização de sensores adicionais, como acelerômetros, sensores de ingestão alimentar e dispositivos de monitoramento da temperatura corporal, poderia fornecer informações mais detalhadas sobre a saúde e o bem-estar das vacas, tornando a análise mais precisa e robusta.

No que se refere à aplicabilidade prática, a acurácia de 77% obtida pelo modelo *LSTM* é considerada satisfatória para ser utilizada como um primeiro nível de triagem. O modelo pode funcionar como um sistema de alerta inicial, identificando mudanças sutis no comportamento das vacas que possam indicar o desenvolvimento de alguma enfermidade. Dessa forma, a ferramenta se torna um suporte valioso para os produtores, auxiliando na tomada de decisões e permitindo a implementação de medidas preventivas antes que o quadro clínico dos animais se agrave.

Para viabilizar a utilização desse modelo em um cenário real, sugere-se a implementação de um sistema contínuo de monitoramento, no qual os dados coletados diariamente sejam inseridos no modelo *LSTM* já treinado. Esse sistema pode incluir informações sobre movimentação, tempo de descanso, padrões de locomoção e ingestão alimentar, que seriam analisadas para detectar variações anormais. Ao identificar comportamentos atípicos, o modelo poderia gerar alertas automáticos, permitindo que os produtores investiguem possíveis causas e adotem intervenções adequadas para evitar prejuízos à saúde do rebanho. Além disso, a integração do modelo com sistemas de gestão agropecuária e dispositivos *Iot* pode otimizar ainda mais sua eficácia, garantindo um monitoramento mais preciso e automatizado.

Por fim, como sugestão para trabalhos futuros, recomenda-se o desenvolvimento de metodologias capazes de identificar padrões específicos para diferentes enfermidades. A ampliação da base de dados com informações detalhadas sobre doenças comuns no rebanho leiteiro pode contribuir para o treinamento de modelos mais especializados, tornando o sistema não apenas capaz de detectar possíveis problemas de saúde, mas também de sugerir diagnósticos mais precisos. Dessa forma, o uso da inteligência artificial na pecuária leiteira poderá ser ainda mais eficiente, auxiliando na redução de perdas produtivas e na melhoria da qualidade de vida dos animais.

Referências

ARAÚJO, W. dos S.; SOUZA, F. d. A. S.; BRITO, J. I. B. de; LIMA, L. M. Aplicação do modelo estocástico cadeia de markov a dados diários de precipitação dos estados da bahia e sergipe. **Revista Brasileira de Geografia Física**, v. 3, p. 509–523, 2012.

BENAISSA, S.; TUYTTENS, F.; PLETS, D.; MARTENS, L.; VANDAELE, L.; JOSEPH, W.; SONCK, B. Improved cattle behaviour monitoring by combining ultra-wideband location and accelerometer data. **animal**, v. 17, n. 4, p. 100730, 2023.

BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [S.l.]: Novatec Editora, 2014.

CAMPOS, F. S.; ASSIS, F. A.; SILVA, A. M. L. da; COELHO, A. J.; MOURA, R. A.; SCHROEDER, M. A. O. Reliability evaluation of composite generation and transmission systems via binary logistic regression and parallel processing. **International Journal of Electrical Power & Energy Systems**, Elsevier, v. 142, p. 108380, 2022.

CHURAKOV, M.; SILVERA, A. M.; GUSSMANN, M.; NIELSEN, P. P. Parity and days in milk affect cubicle occupancy in dairy cows. **Applied Animal Behaviour Science**, v. 244, p. 105494, 2021.

DAVIS, L.; FRENCH, E. A.; AGUERRE, M. J.; ALI, A. Impact of parity on cow stress, behavior, and production at a farm with guided traffic automatic milking system. **Frontiers in Animal Science**, v. 4, 2023.

GUIEN, V.; ANTOINE, V.; LARDY, R.; MOREIRA, I. da R.; ROCHA, L. E. C.; VEISSIER, I. Simulation de comportement d'une vache à l'aide d'un processus markovien. 2023.

GURUMURTHY, B.; JAYALAKSHMI, S. Predictive scaling for elastic compute resources on public cloud utilizing deep learning based long short-term memory. **International Journal of Advanced Computer Science and Applications**, v. 12, 01 2021.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. *et al.* **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

LARDY, R.; MIALON, M.-M.; WAGNER, N.; GAUDRON, Y.; MEUNIER, B.; SLOTH, K. H.; LEDOUX, D.; SILBERBERG, M.; ROCHES, A. d. B. des; RUIN, Q. *et al.* Understanding anomalies in animal behaviour: data on cow activity in relation to health and welfare. **Animal-Open Space**, Elsevier, v. 1, n. 1, p. 5, 2022.

LARDY, R.; RUIN, Q.; VEISSIER, I. Discriminating pathological, reproductive or stress conditions in cows using machine learning on sensor-based activity data. **Computers and Electronics in Agriculture**, v. 204, p. 107556, 2023. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016816992200864X>.

LI, L.; GUO, D.; SHI, C.; ZHENG, Y. The predictive role of sedentary behavior and physical activity on adolescent depressive symptoms: A machine learning approach. **Journal of Affective Disorders**, v. 378, p. 81–89, 2025. ISSN 0165-0327. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0165032725003027>.

LIANG, H.-T.; HSU, S.-W.; HSU, J.-T.; TU, C.-J.; CHANG, Y.-C.; JIAN, C. T.; LIN, T.-T. An imu-based machine learning approach for daily behavior pattern recognition in dairy cows. **Smart Agricultural Technology**, v. 9, p. 100539, 2024. ISSN 2772-3755. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2772375524001448>.

LUO, J.; ZHU, D.; LI, D. Classification-enhanced lstm model for predicting river water levels. **Journal of Hydrology**, v. 650, p. 132535, 2025. ISSN 0022-1694. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022169424019310>.

MATTACHINI, G.; RIVA, E.; PROVOLO, G. The lying and standing activity indices of dairy cows in free-stall housing. **Applied Animal Behaviour Science**, v. 129, n. 1, p. 18–27, 2011.

MOREIRA, I. da R. **Simulation de vaches numériques**. 2023. Rapport d'élève ingénieur, Stage de 2e année, Filière : Systèmes d'Information et Aide à la Décision. Encadrante: Violaine ANTOINE.

PRADO, D. **Teoria das Filas e da Simulação**. [S.l.]: Falconi Editora, 2022. v. 2.

ROTHE, F.; LAMES, M. Simulation of tennis behaviour using finite markov chains. **IFAC-PapersOnLine**, v. 55, n. 20, p. 606–611, 2022. ISSN 2405-8963. 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896322013672>.

SILBERBERG, M.; MIALON, M.; MEUNIER, B.; VEISSIER, I. Sensor-captured modifications in cow behaviour under subacute ruminal acidosis. **Animal - Open Space**, v. 3, p. 100063, 2024.

SILVA, E. M. M. Aplicação da simulação baseada em agentes no desenvolvimento de uma ferramenta de ensino baseada em simulação. IEPG-Instituto de Engenharia de Produção e Gestão, 2017.

SOUZA, G. P. Cadeias de markov. **Recuperado em 23/05/2023, de https://pt.scribd.com/doc/52238456/Tutorial-Cadeias-de-Markov**, v. 19, n. 10, p. 8, 2011.

TULLO, E.; FONTANA, I.; GOTTARDO, D.; SLOTH, K.; GUARINO, M. Technical note: Validation of a commercial system for the continuous and automated monitoring of dairy cow activity. **Journal of Dairy Science**, v. 99, n. 9, p. 7489–7494, 2016.

VALENTIN, G. Rapport d'élève ingénieur stage de 3e année: Recalage de séries temporelles de rythmes d'activité de vaches laitières. **LIMOS**, p. 58, 2022.

VIDAL, G.; SHARPNACK, J.; PINEDO, P.; TSAI, I. C.; LEE, A. R.; MARTíNEZ-LóPEZ, B. Impact of sensor data pre-processing strategies and selection of machine learning algorithm on the prediction of metritis events in dairy cattle. **Preventive Veterinary Medicine**, v. 215, p. 105903, 2023. ISSN 0167-5877. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167587723000673>.

VOS, K.; PENG, Z.; JENKINS, C.; SHAHRIAR, M. R.; BORGHESANI, P.; WANG, W. Vibration-based anomaly detection using lstm/svm approaches. **Mechanical Systems and Signal Processing**, v. 169, p. 108752, 2022. ISSN 0888-3270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0888327021010682>.

WAGNER, N. Patterns d'organisation circadienne des activités des animaux en relation avec les états pré-pathologiques. **LIMOS**, p. 135, 2020.

WAGNER, N.; ANTOINE, V.; MIALON, M.-M.; LARDY, R.; SILBERBERG, M.; KOKO, J.; VEISSIER, I. Machine learning to detect behavioural anomalies in dairy cows under subacute ruminal acidosis. **Computers and electronics in agriculture**, Elsevier, v. 170, p. 105233, 2020.

WAGNER, N.; MIALON, M.-M.; SLOTH, K. H.; LARDY, R.; LEDOUX, D.; SILBERBERG, M.; ROCHES, A. d. B. D.; VEISSIER, I. Detection of changes in the circadian rhythm of cattle in relation to disease, stress, and reproductive events. **Methods**, Elsevier, v. 186, p. 14–21, 2021.

YU, L.; YANG, X.; WEI, H.; LIU, J.; LI, B. Driver fatigue detection using ppg signal, facial features, head postures with an lstm model. **Heliyon**, v. 10, n. 21, p. e39479, 2024. ISSN 2405-8440. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405844024155107>.

ZHENG, M.; LUO, X. Joint estimation of state of charge (soc) and state of health (soh) for lithium ion batteries using support vector machine (svm), convolutional neural network (cnn) and long sort term memory network (lstm) models. **International Journal of Electrochemical Science**, v. 19, n. 9, p. 100747, 2024. ISSN 1452-3981. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1452398124002888>.

ZHOU, P.; WANG, H.; LI, F.; DAI, Y.; HUANG, C. Development of window opening models for residential building in hot summer and cold winter climate zone of china. **Energy and Built Environment**, v. 3, n. 3, p. 363–372, 2022.