INSTITUTO FEDERAL DE SÃO PAULO – IFSP

CAMPUS CATANDUVA

TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Detecção de Minas Terrestres: uma abordagem com inteligência artificial

GABRIEL GOMES DA SILVA
RENNER ANTUNES PRADO

Catanduva - SP

13 de dezembro de 2023

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecária: Luiza Correia Lima Felix – CRB-8/10837

Prado, Renner Antunes.
P896d Detecção de Minas Terrestres: uma abordagem com inteligência artificial / Renner Antunes Prado, Gabriel Gomes da Silva.
Catanduva, SP : IFSP, 2023.
43 f. : il.

Trabalho de Conclusão de Curso (Tecnólogo – Análise e Desenvolvimento de Sistema) – Instituto Federal de São Paulo, campus Catanduva, 2023.
Orientador: Prof. Me. Flávio Luiz dos Santos de Souza.

1. Detecção de Minas Terrestres. 2. Inteligência Artificial.
3. Machine Learning. 4. Recall.
I. Silva, Gabriel Gomes da. II. Título.

CDD (23 ed.): 004.8

ATA N.º 5/2023 - SIF-CTD/DAE-CTD/DRG/CTD/IFSP

Ata de Defesa de Trabalho de Conclusão de Curso - Tecnologia em Análise e Desenvolvimento de Sistemas

Na presente data realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado **Detecção de Minas Terrestres: uma abordagem com inteligência artificial** apresentado pelo aluno **Gabriel Gomes da Silva (CT3013723)** e **Renner Antunes Prado (CT3009475)** do Curso Lato Senso de **TECNOLOGIA EM ANALISE E DESENVOLVIMENTO DE SISTEMAS,** (Câmpus Campus Catanduva). Os trabalhos foram iniciados às 19:20 pelo(a) Professor(a) presidente da banca examinadora, constituída pelos seguintes membros:

Membros	IES	Presença (Sim/Não)	Aprovação/Conceito (Quando Exigido)
Flavio Luiz dos Santos de Souza (Presidente/Orientador)	IFSP-CTD	Sim	APROVADO/10.0
Fabio Luiz Viana (Examinador 1)	IFSP-CTD	Sim	APROVADO/10.0
Márcio Andrey Teixeira (Examinador 2)	IFSP-CTD	Sim	APROVADO/10.0

Observações:

A banca examinadora, tendo terminado a apresentação do conteúdo da monografia, passou à arguição do(a) candidato(a). Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo(a) aluno(a), tendo sido atribuído o seguinte resultado:

[X] Aprovado(a) [ ] Reprovado(a) Nota (quando exigido): 10,0

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu lavrei a presente ata que assino juntamente com os demais membros da banca examinadora.

Câmpus Catanduva, 28 de novembro de 2023

Avaliador externo: [ ]Sim [X]Não

Assinatura:

Documento assinado eletronicamente por:

- **Flavio Luiz dos Santos de Souza**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/11/2023 07:38:12.
- **Marcio Andrey Teixeira**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/11/2023 14:04:29.
- **Fabio Luiz Viana**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/11/2023 15:22:01.
- **Gabriel Gomes da Silva**, **CT3013723 - Discente**, em 29/11/2023 15:31:51.
- **RENNER ANTUNES PRADO**, **CT3009475 - Discente**, em 29/11/2023 15:44:03.

Este documento foi emitido pelo SUAP em 29/11/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifsp.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 655225
Código de Autenticação: d293f95a85

Dedicamos este trabalho à Deus, aos nossos pais, familiares e amigos.

AGRADECIMENTOS

Nós desejamos expressar nossa gratidão a Deus pela força que nos proporcionou para o desenvolvimento deste trabalho. Gostaríamos de estender nossos agradecimentos aos nossos pais, familiares e amigos por nos incentivarem e acreditarem em nossos esforços ao longo deste curso.

Além disso, queremos agradecer profundamente ao nosso orientador, Prof. Flávio Luiz dos Santos de Souza, Msc. pela orientação e auxílio durante a condução deste trabalho.

"A criação bem-sucedida de inteligência artificial seria o maior evento na história da humanidade. Infelizmente, pode também ser o último, a menos que aprendamos a evitar os riscos."

(Stephen Hawking)

RESUMO

No mundo de hoje, milhões de vidas estão em risco devido à presença persistente de minas terrestres, tornando a detecção eficaz desses artefatos explosivos mais crucial do que nunca. Este trabalho apresenta uma abordagem baseada em técnicas de Inteligência Artificial e *Machine Learning* para a detecção eficaz de minas terrestres. Utilizando o algoritmo `Random Forest` e o conjunto de dados `Land Mines Detection`, exploramos a classificação de áreas contaminadas, visando reduzir riscos humanitários associados a esses artefatos explosivos. O modelo treinado alcançou uma média de 94,00% de Acurácia e um *Recall* de 91,40%. Embora as métricas indiquem um desempenho geral satisfatório, destaca-se a necessidade de equilibrar o *Recall* para evitar riscos de Falsos Negativos em áreas críticas. A combinação de técnicas avançadas de Inteligência Artificial e *Machine Learning* oferece uma abordagem promissora para mitigar as ameaças persistentes representadas por minas terrestres.

Palavras-chave: Detecção de Minas Terrestres, Inteligência Artificial, *Machine Learning*, `Random Forest`, *Recall*.

ABSTRACT

In today's world, millions of lives are at risk due to the persistent presence of landmines, which makes the effective detection of these explosive devices more crucial than ever. This paper presents an approach based on Artificial Intelligence and Machine Learning techniques for the effective detection of landmines. Using the `Random Forest` algorithm and the `Land Mines Detection` dataset, we explored the classification of contaminated areas in order to reduce the humanitarian risks associated with these explosive devices. The trained model achieved an average Accuracy of 94.00% and a Recall of 91.40%. Although the metrics indicate a satisfactory overall performance, there is a need to balance the Recall to avoid the risk of False Negatives in critical areas. The combination of advanced Artificial Intelligence and Machine Learning techniques offers a promising approach to mitigating the persistent threats posed by landmines.

Keywords: Landmine Detection, Artificial Intelligence, Machine Learning, `Random Forest`, Recall.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE CÓDIGOS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

No mundo de hoje, milhões de vidas estão em risco devido à presença persistente de minas terrestres, tornando a detecção eficaz desses artefatos explosivos mais crucial do que nunca. Ao longo da história da humanidade, conflitos armados entre nações sempre resultaram em trágicas consequências. As guerras contemporâneas geraram as minas terrestres, artefatos explosivos projetados para prejudicar inimigos, cujos efeitos persistem muito tempo após o fim dos conflitos. Infelizmente, essas minas ainda ameaçam pessoas inocentes, causando perdas de vidas e bens materiais. Segundo a ONU (2022), somente em 2021, 5.544 vítimas de minas terrestres foram registradas, sendo 2.182 delas fatais.

Para enfrentar esse desafio crítico, este trabalho adota técnicas de Inteligência Artificial e *Machine Learning* (Aprendizado de Máquina). Essas técnicas permitem o uso de dados coletados em áreas minadas para treinar algoritmos capazes de identificar a presença de minas terrestres. Neste trabalho, foi empregado o aprendizado supervisionado por meio do algoritmo de *Machine Learning* `Random Forrest`, que permite a classificação dos dados em uma classe de mina resultante. Trabalhos anteriores, como Pryshchenko et al. (2022), demonstraram o potencial da Inteligência Artificial na detecção de minas utilizando de Redes Neurais e dados coletados por antenas que captam sinais do solo, obtendo resultados consistentes e precisos. Outros trabalhos, como Makki et al. (2018), Achkar, Owayjan e Mrad (2011), também exploraram abordagens baseadas em *Machine Learning* para detectar minas terrestres. Além disso, existem trabalhos que utilizaram o mesmo conjunto de dados empregado neste trabalho. Um deles, apresentado por Pradana (2023), obteve um resultado de 83,82% de Acurácia, indicando que esse tipo de abordagem é promissora.

Neste projeto, foi utilizado o conjunto de dados (`Dataset`) `Land Mines Detection`[1], que fornece informações obtidas por sensores *Fluxgate Magnetometer* (FLC) capazes de detectar ondas eletromagnéticas do solo, indicando a presença de minas. Para alcançar resultados que sejam considerados elevados foi necessário realizar o processamento, limpeza e balanceamento dos dados, e a seleção dos algoritmos apropriados para o treinamento do modelo.

O objetivo principal do presente trabalho é desenvolver um modelo de *Machine Learning* capaz de identificar a presença ou ausência de minas terrestres em áreas contaminadas. Esperamos que este estudo demonstre que a combinação de técnicas de Inteligência Artificial e *Machine Learning* oferece um meio promissor para mitigar as ameaças persistentes representadas por esses artefatos explosivos, reduzindo a perda de bens materiais e,

[1] <https://www.kaggle.com/datasets/ritwikb3/land-mines-detection>

principalmente, o risco para a vida humana.

1.1 Objetivos

Auxiliar países que têm regiões afetadas por minas terrestres a realizar a desminagem com maior eficiência e menor perda de recursos humanos. Esse é um dos problemas em que a tecnologia e a Inteligência Artificial podem ajudar a melhorar a vida das pessoas.

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é treinar um modelo de *Machine Learning* capaz de realizar a detecção de minas terrestres de forma eficiente e precisa, visando reduzir o risco de acidentes e contribuir para a segurança em áreas afetadas por minas terrestres.

1.1.2 Objetivos Específicos

O principal objetivo desta pesquisa é facilitar a identificação e remoção de minas terrestres em áreas suscetíveis, fornecendo então uma Inteligência Artificial com a capacidade de detectar estes artefatos explosivos. Para o projeto ser desenvolvido, serão necessários os seguintes objetivos específicos:

- Estudar e explicar modelos e algoritmos de *Machine Learning* para a análise e classificação de dados referentes às minas terrestres.
- Fazer a coleta de `Datasets`, cujo conteúdo visa características a cerca de solos com a presença de minas terrestres ou não.
- Fazer o uso e comparação de métricas de *Machine Learning*.
- Realizar o treinamento de um modelo de *Machine Learning* capaz de identificar a presença de minas terrestres em solos.

1.2 Justificativa

Segundo Waterhouse (2023) acredita-se que cerca de 174 mil quilômetros quadrados de terras na Ucrânia estejam contaminadas por minas terrestres. Essa extensão de território equivale a quatro vezes o tamanho do estado do Rio de Janeiro. Na região de Kharkiv mais de 55 mil explosivos já foram detectados desde setembro de 2022.

Metade das vítimas fatais causadas por esses explosivos em todo o mundo nos últimos sete anos são crianças (ONU, 2022). Diante desse sério problema deixado pelas guerras,

existem pessoas corajosas que trabalham incessantemente para mapear e descontaminar áreas infestadas por minas terrestres.

De acordo com Foster (2022), somente no Líbano em 2022, estimava-se que grupos encarregados pela limpeza dos solos teriam desminado dois milhões de metros quadrados de terra e desativado cerca de 10 mil minas terrestres. Graças a estes grupos de pessoas que trabalham limpando regiões afetadas pelas minas terrestres, é possível obter dados vitais acerca do processo, como dados de solos, voltagens mostradas no equipamento, tipos de minas encontradas e entre outros.

O processo de detecção e remoção de minas terrestres pode ser extremamente demorado. Como ainda não existe uma técnica ou método definitivo que seja eficiente e preciso o suficiente, essa tarefa se torna bastante perigosa, mesmo para pessoas com treinamento adequado (ANDRAOS; MARANHÃO; GUEDES, 2018). A cada dia que passa, mais pessoas são recrutadas para realizar a detecção de minas terrestres, aumentando a necessidade contínua de aprimorar e desenvolver novas técnicas e métodos para auxiliar na detecção, visando diminuir o número de Falsos Negativos e, assim, protegendo vidas e bens.

1.3 Metodologia

Primeiramente, ao pensar na solução deste problema, é necessário ter um bom conjunto de dados para realizar o treinamento. Por causa disso, o primeiro passo foi a escolha do conjunto de dados a ser utilizado. Com o `Dataset` em mãos, foi feita a análise de como os dados se comportam e suas correlações. Através dessa análise, pôde-se chegar à conclusão sobre quais modelos deveriam ser utilizados para o treinamento do modelo de *Machine Learning*.

Após a escolha do `Dataset`, realizou-se a análise e o processamento dos dados, fazendo os ajustes necessários para o treinamento. Nesse processo, foi feito a Análise e Exploração de dados seguida de Pre-processamento de dados do `Dataset`.

Terminado o processo de análise e processamento dos dados, chegou o momento de partir para o planejamento para o treinamento. Assim, se faz necessário selecionar algoritmos de *Machine Learning* como a validação cruzada e a técnica de grade para encontrar os melhores hiper-parâmetros para otimizar os resultados do modelo. No ultimo passo foi executado o Treinamento Avaliativo dos modelos com os hiper-parâmetros encontrados anteriormente.

1.4 Trabalhos Relacionados

A detecção de minas terrestres é uma questão importante devido aos riscos humanitários associados. Ao longo dos anos, uma variedade de abordagens tem sido exploradas para lidar com esse desafio, desde métodos convencionais até o emprego de tecnologias mais avançadas, incluindo a Inteligência Artificial. Nesta seção, serão revisados alguns trabalhos relacionados, abordando tanto métodos convencionais quanto aplicações de Inteligência Artificial na detecção de minas terrestres.

1.4.1 Detecção de Minas Terrestres: Abordagens Convencionais

Existem diversas técnicas e abordagens para realizar a detecção de minas, nesta seção serão abordadas algumas técnicas consideradas convencionais na detecção de minas terrestres.

Um dos métodos mais básicos que podem ser utilizados é a Inspeção Visual. Segundo Andraos, Maranhão e Guedes (2018), a Inspeção Visual é uma abordagem simples que pode ser usada a fim de detectar artefatos explosivos, neste método é observado a composição do terreno com o intuito de identificar objetos semelhantes a minas terrestres, porém devido ao fato de ser simples, este método é limitado e está sujeito a falhas humanas.

Outra técnica de detecção utilizada é a Detecção de Metal. Ainda de acordo com Andraos, Maranhão e Guedes (2018), o método de Detecção de Metal é executado a partir do conceito de indução eletromagnética, onde são criados campos eletromagnéticas por meio de bobinas presentes no equipamento. Através dos campos eletromagnéticos uma corrente `Foucault` é induzida nos objetos metálicos e uma corrente oposta é gerada no detector, o equipamento detecta esta corrente, e consecutivamente, a mina terrestre.

Uma abordagem interessante é a de Animais Adestrados para farejarem a presença de minas terrestres. Segundo Nunes (2005), Animais Adestrados costuma ser executada por cães, pois eles têm seu olfato mais apurado em relação aos humanos e são mais eficientes na detecção da presença de explosivos. A partir dos vapores emanados pelos objetos explosivos os cães conseguem os farejar. Entretanto os animais utilizados neste método possuem uma estimativa de vida relativamente curta, o que dificulta a continuidade do processo.

O método de *Ground-Penetrating Radar* (GPR) é uma abordagem promissora para a detecção de minas terrestres, emitindo ondas de rádio no solo e analisando os sinais de retorno para identificar objetos enterrados. Conforme MacDonald et al. (2003), a GPR utiliza antenas para emitir e captar sinais, interpretando-os computacionalmente para gerar uma imagem visual ou um sinal de áudio indicando semelhança com minas. No entanto, apresenta limitações em relação a sensibilidade a irregularidades naturais, necessidade de equilíbrio entre resolução e profundidade, e a complexidade na interação entre conteúdo metálico, frequência de interrogatório e características do solo.

1.4.2 Aplicações de Inteligência Artificial na Detecção de Minas

Nesta seção, será conduzida uma análise de trabalhos correlatos que se alinham tematicamente ao escopo deste estudo, centrado na detecção de minas terrestres com o apoio de técnicas de Inteligência Artificial. Na Tabela 1, apresentamos um pequeno comparativo entre os objetivos, técnicas e resultados dos trabalhos correlatos.

Tabela 1 – Comparativo entre os trabalhos correlatos.

Trabalho	**Objetivo**	**Técnicas**	**Resultados**
Implementation of an Artificial Intelligence Approach to GPR Systems for Landmine Detection (PRYSHCHENKO et al., 2022)	Detecção de minas terrestres utilizando de Redes Neurais Artificiais.	Redes Neurais, *Ground Penetrating Radar*, análise de sinais eletromagnéticos.	Acurácia de validação de 90%, boa detecção de objetos subsuperficiais.
RBF Neural Network for Landmine Detection in H Yperspectral Imaging (MAKKI et al., 2018)	Avaliar algoritmos de classificação para detecção de múltiplos alvos em imagens hiperespectrais, com foco na detecção de minas terrestres.	*Radial Basis Function Neural Networks*, imagens hiper-espectrais.	*Radial Basis Function Neural Networks* com Probabilidade de Detecção de 100% e taxa de alarme falso médio igual a zero.
Adaptive Learning Approach to Landmine Detection (SUN; LI, 2005)	Detecção de minas com *Forward-looking ground penetrating radar*.	*Forward-looking ground penetrating radar*, `AdaBoost` e Rede Neural.	Utilizando do `AdaBoost` juntamente com Redes Neurais obtiveram uma taxa mínima de alarmes falsos e uma alta probabilidade de detecção.
GPR Signal Characterization for Automated Landmine and UXO Detection Based on Machine Learning Techniques (NúñEZ-NIETO et al., 2014)	Sistema de detecção em tempo real para a Marinha da Espanha com *Ground Penetrating Radar*.	*Ground Penetrating Radar*, Regressão Logística, Redes Neurais.	Rede Neural mais precisa que Regressão Logística, destaque para antena de 2.3GHz.

O trabalho proposto por Pryshchenko et al. (2022) é uma abordagem de Redes Neurais para a detecção de minas terrestres antipessoais, visando auxiliar na desafiadora tarefa de desminamento nas regiões de Donetsk e Luhansk, na Ucrânia, por meio de

dispositivos controlados remotamente. Essa detecção é realizada por meio da análise de sinais do GPR, que consistem em sinais eletromagnéticos. O estudo evidencia uma preocupação significativa com os Falsos Negativos resultantes da presença de outros objetos enterrados. Na seção de resultados do trabalho, é mencionado que foi obtido uma Acurácia de validação de cerca de 90%, o que é um ótimo desempenho. Os resultados obtidos neste estudo também indicam que a utilização de Redes Neurais para o reconhecimento de objetos utilizando GPR possibilita a detecção confiável de objetos subsuperficiais, mesmo em condições de ruído elevado.

No estudo conduzido por Makki et al. (2018), a proposta consiste na avaliação de algoritmos de classificação para detecção de alvos múltiplos em imagens hiper-espectrais, com ênfase na detecção de minas terrestres. A introdução de métodos baseados em Inteligência Artificial visa aprimorar o desempenho nas etapas de detecção, identificação e estimativa de abundância dos alvos. Destaca-se a eficácia da estratégia de treinamento para as *Radial Basis Function Neural Networks* (RBFNN) em imagens hiper-espectrais, evidenciando sua vantagem na realização de todas as etapas anteriores. A análise comparativa, utilizando métricas como a *Probability of Detection* (PD), a *False Alarm Rate* (FAR) e o *Computation Time* (CT), revela resultados impressionantes. O modelo RBFNN, após treinamento intensivo, demonstrou uma PD de 1, indicando detecção perfeita, associada ao mínimo FAR. Notavelmente, o RBFNN exibiu uma média de FAR igual a zero, indicando a inexistência de classificações de Falsos Positivos. Além disso, o modelo apresentou uma CT de 333,25 segundos para analisar 17 imagens, reforçando seu desempenho computacional satisfatório.

No estudo realizado por Sun e Li (2005) sobre detecção de minas terrestres utilizando um *Forward-looking Ground Penetrating Radar* (FLGPR), foram enfrentados desafios relacionados à extração de estruturas complexas dos sinais-alvo e à adaptação do classificador ao ambiente ao redor. A Seleção Sequencial de Avanço Flutuante foi empregada para extrair componentes relevantes para um classificador de Rede Neural. Para lidar com a detecção de minas em ambientes desafiadores, incorporam o algoritmo `AdaBoost`, integrando a seleção de características em cada iteração. Na seção de resultados experimentais o trabalho foca mais em métricas qualitativas, como por exemplo a curva ROC, que é mencionada sendo empurrada para o canto superior esquerdo, sugerindo um bom desempenho. Por fim os resultados apontaram melhorias significativas no desempenho de classificação, destacando a potencial extensão dessa abordagem para problemas gerais de reconhecimento de objetos.

Um trabalho desenvolvido por Núñez-Nieto et al. (2014), teve como objetivo principal o desenvolvimento um sistema de detecção de objetos não detonados em tempo real para a Marinha do Corpo de Fuzileiros Navais da Espanha. O trabalho utilizou um GPR para coletar informações sobre um solo arenoso, foram coletados sinais captados

do solo utilizando antenas de 1 e 2.3GHz. No treinamento utilizou-se de dois algoritmos diferentes, Regressão Logística e Redes Neurais. Ao comparar os resultados obtidos, chegou-se a conclusão que a Rede Neural se destacou da Regressão Logística, sendo mais precisa especialmente em cenários que apresentavam grande número de ruídos. Foram observados os resultados com base na Acurácia e chegou-se a conclusão que os resultados obtidos pela antena de maior frequência (2.3GHz), são mais precisas do que os com a antena de menor frequência. Ao final o algoritmo de Regressão Logística obteve 57% Acurácia, 28% de Falsos Positivos e 15% de Falsos Negativos com a antena de 1GHz e 65% de Acurácia, 17% de Falsos Positivos e 18% de Falsos Negativos com a antena de 2.3Ghz. Já a Rede Neural obteve 89% de Acurácia, 7% de Falsos Positivos e 4% de Falsos Negativos com a antena de 1GHz e 92% de Acurácia, 4% de Falsos Positivos e 4% de Falsos Negativos com a antena de 2.3GHz.

2 DESENVOLVIMENTO

2.1 Ferramentas e Tecnologias

A Tabela 2 lista as ferramentas e tecnologias que foram utilizadas no desenvolvimento deste projeto, juntamente com suas versões correspondentes e links de download. Cada uma dessas ferramentas desempenhou um papel no decorrer dos processos de análise, pré-processamento de dados, treinamento de modelos e avaliação de desempenho. As principais ferramentas serão descritas nas seções 2.1.1, 2.1.2 e 2.1.3.

Tabela 2 – Ferramentas e Tecnologias utilizadas

Nome	**Versão**	**Download**	**Etapa de Uso**
Python	3.11.5	<https://www.python.org/downloads/release/python-3115/>	Análise, pré-processamento de dados, treinamento de modelos e avaliação de desempenho
Scikit-learn	1.2.2	<https://scikit-learn.org/stable/install.html>	Treinamento de modelos e avaliação de desempenho
Jupyter Notebook	6.5.4	<https://jupyter.org/install>	Análise, pré-processamento de dados, treinamento de modelos e avaliação de desempenho
Pandas	2.0.2	<https://pandas.pydata.org/getting_started.html>	Análise, pré-processamento de dados
Numpy	1.24.3	<https://numpy.org/install/>	Análise e Avaliação de desempenho
Seaborn	0.13.0	<https://seaborn.pydata.org/installing.html>	Análise e Avaliação de desempenho
Plotly	5.15.0	<https://plotly.com/python/getting-started/>	Análise
Imblearn	0.10.1	<https://imbalanced-learn.org/stable/install.html>	Pré-processamento de dados

2.1.1 Python

Segundo Kriger (2023), o `Python` é uma linguagem de programação desenvolvida no início dos anos noventa pelo matemático holandês Guido Van Rossum, ela surgiu da necessidade de economizar tempo e melhorar a eficiência no desenvolvimento de projetos. Desde a sua concepção, o `Python` foi pensado para ser uma linguagem simples e poderosa que possibilita construir desde *Scripts* simples até sistemas extremamente complexos. Por causa de sua otimização de leitura de código que estimula a produtividade, diversos profissionais das mais variadas áreas começaram a adotar a linguagem (KRIGER, 2023). O logo dessa linguagem pode ser observada na Figura 1[1].

[1] <https://www.python.org/>

Neste trabalho, a linguagem de programação `Python` foi utilizado como linguagem principal para o desenvolvimento da Inteligência Artificial. Foram utilizadas diversas bibliotecas e *Frameworks* presentes na linguagem para realizar desde a análise, exploração e manipulação de dados até o treinamento e teste dos modelos de *Machine Learning* descritos nos objetivos específicos.

Figura 1 – Logo Python

2.1.2 Scikit-learn

De acordo com Marcela (2023), `Scikit-Learn` é um *Framework Open Source* da linguagem `Python` para o desenvolvimento de aplicações de *Machine Learning* que fornece diversos recursos para a modelagem estatística, análise e mineração de dados, além de oferecer suporte a algoritmos de aprendizado supervisionado e não supervisionado. Na Figura 2[2] é apresentada sua logo.

A biblioteca surgiu como um projeto do Google *Summer of Code*, que foi idealizado pelo cientista de dados David Cournapeau. A mesma é considerada uma das soluções mais versáteis, populares e robustas do mercado atualmente, pois em seu código fonte é empregado o uso de outras bibliotecas numéricas e científicas já consolidadas na comunidade, como o `NumPy` e `Matplotlib` (MARCELA, 2023).

O `Scikit-Learn` desempenha um papel fundamental no presente trabalho, pois nele estão implementados diversos modelos de *Machine Learning* que foram utilizados e testados no desenvolvimento desta pesquisa, como algoritmos para realizar a medição dos resultados dos modelos e a separação dos conjuntos de dados para treino e teste.

Figura 2 – Logo Scikit-Learn

2.1.3 Jupyter Notebook

De acordo com Miranda (2021), `Jupyter Notebook` é uma *Integrated Development Environment* (IDE) (Ambiente de desenvolvimento integrado) que oferece todas as funcio-

[2] <https://scikit-learn.org/stable/>

nalidades de um notebook virtual. Um notebook é um ambiente em que se pode escrever histórias de forma a se seguir um fluxo. Esses fluxos podem ser divididos em blocos com códigos, instruções ou textos, esses blocos são denominados células. A Partir das células de código é possível fazer a exploração e tratamento de conjuntos de dados e até mesmo plotar gráficos.

Miranda (2021) define a utilidade das células da seguinte forma:

> Através dos blocos de texto, podemos explicar o contexto, o objetivo do nosso projeto, o conhecimento que está sendo extraído dos dados e as conclusões, ou seja, as possíveis soluções para o problema que estamos tentando resolver, ou até mesmo se ainda não conseguimos chegar a nenhuma solução.

O `Jupyter Notebook` é amplamente utilizado devido a sua interface amigável e facilidade para criar protótipos de forma rápida, também é muito utilizada para trabalhar com a ciência de dados, que tem uma grande influência na área de *Machine Learning* a qual nosso trabalho se sustenta. É possível observar o logo do `Jupyter Notebook` na Figura 4[3]. Por todos estes motivos o `Jupyter Notebook` foi utilizado como o ambiente de desenvolvimento do presente estudo. Na Figura 3 temos uma breve introdução à interface visual do `Jupyter Notebook`.

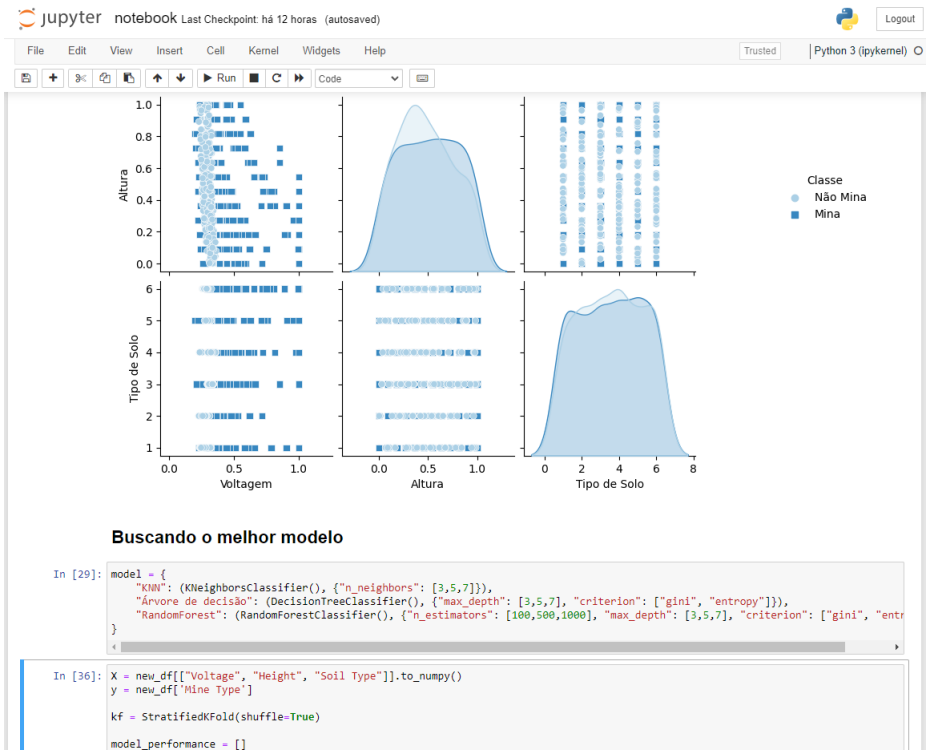


Figura 3 – Interface do ambiente de desenvolvimento `Jupyter Notebook`

[3] <https://jupyter.org/>

Figura 4 – Logo Jupyter Notebook

2.2 Descrição do Dataset

Antes de proceder com o treinamento do modelo de detecção de minas terrestres, se apresentou necessário a obtenção de um conjunto de dados que fosse coeso e coerente com os ideais do trabalho. O `Dataset` utilizado para este trabalho não foi produzido internamente, mas sim adquirido de fontes externas, portanto, o pré-processamento de desempenha um papel crítico na garantia da qualidade e na preparação dos dados para análise.

O conjunto de dados escolhido para o desenvolvimento do presente trabalho é o `Land Mines Detection`, obtido da plataforma Kaggle[4]. Este `Dataset` contém 338 amostras de solos diferentes, utilizando um sensor FLC para realizar as medições de campos eletromagnéticos. O conjunto de dados é dividido em cinco rótulos para a classificação de minas terrestres. É possível observar na Figura 5 algumas das amostras contidas no `Dataset`.

	Voltagem	Altura	Tipo de Solo	Tipo de Mina
326	0.446284	0.000000	5	5
63	0.719032	0.000000	2	2
27	0.302114	0.454545	5	1
251	0.586102	0.727273	1	2
171	0.233988	0.727273	3	4
143	0.368580	0.818182	1	4
303	0.371601	0.636364	2	4
210	0.295347	1.000000	5	5
320	0.540785	0.090909	4	5
283	0.480362	0.090909	5	3

Figura 5 – Amostras de dados do `Dataset`

Para cada registro, o `Dataset Land Mines Detection` demonstra três atributos de características e um identificador de classe. Dos três atributos, dois são quantitativos, o ultimo atributo junto do identificador são qualitativos. Os atributos do `Dataset`, conforme mostrado na Tabela 3 e na Figura 5 são: Voltagem (*Voltage*), Altura (*Height*), Tipo de

[4] <https://www.kaggle.com>

Solo (*Soil Type*), Tipo de Mina (*Mine Type*). Dentre esses atributos, o "Tipo de Solo" é dividido em seis categorias:

1. Seco e Arenoso (*Dry and Sandy*)
2. Seco e Húmus (*Dry and Humus*)
3. Seco e Calcário (*Dry and Limy*)
4. Úmido e Arenoso (*Humid and Sandy*)
5. Úmido e Húmus (*Humid and Humus*)
6. Úmido e Calcário (*Humid and Limy*)

As instâncias do `Dataset` podem ter um dos seguintes identificadores contidos no atributo "Tipo de Mina":

1. Nulo (*Null*)
2. Antitanque (*Anti-Tank*)
3. Armadilha Antipessoal (*Booby Trapped Anti-personnel*)
4. Antipessoal (*Anti-personnel*)
5. M14 Antipessoal (*M14 Anti-personnel*)

Tabela 3 – Atributos do `Dataset Land Mines Detection`.

No.	**Atributo**	**Tipo**	**Descrição**
1	Voltagem (*Voltage*)	Quantitativo	Valor da tensão de saída do sensor FLC devido à distorção magnética
2	Altura (*Height*)	Quantitativo	Altura do sensor em relação ao solo
3	Tipo de Solo (*Soil Type*)	Qualitativo	Tipo de solo em que a mina foi encontrada
4	Tipo de Mina (*Mine Type*)	Qualitativo	Tipo de mina que foi encontrada

2.3 Pré-processamento de Dados

No âmbito deste projeto, o objetivo central reside na detecção de minas terrestres presentes no solo. Durante a análise dos dados, surgiu a necessidade de simplificar o conjunto de classes que representam diferentes tipos de minas com o intuito de otimizar o modelo e obter melhores resultados. A transformação aplicada consistiu na combinação dos rótulos 2, 3, 4 e 5 em um único, que passou a ser denominada "Mina". Como resultado dessa transformação, os rótulos foram reconfigurados da seguinte maneira: 0 - "Não mina" e

1 - “Mina”. Essa adaptação visa fornecer uma abordagem mais precisa e eficaz na detecção da presença de minas terrestres, ao mesmo tempo em que simplifica a complexidade do modelo, focando no objetivo fundamental de identificação.

Após a transformação, as classes que eram inicialmente consideradas balanceadas, conforme a Figura 6, tornaram-se severamente desbalanceadas, com uma representação de 21% para “Não Mina” e 79% para “Mina”. Essa discrepância pode ser observada com mais detalhes na Figura 7a. Em conjuntos de dados desequilibrados, onde o número de instâncias de uma classe é muito menor do que o número de instâncias das outras, tende-se a surgir um problema. Classes menores, frequentemente contendo dados importantes para a classificação, são muitas vezes ignoradas pelos classificadores, criando um viés em favor das classes majoritárias (RAMYACHITRA; MANIKANDAN, 2014). Para evitar esse cenário, foi necessário realizar um balanceamento no `Dataset` transformado.

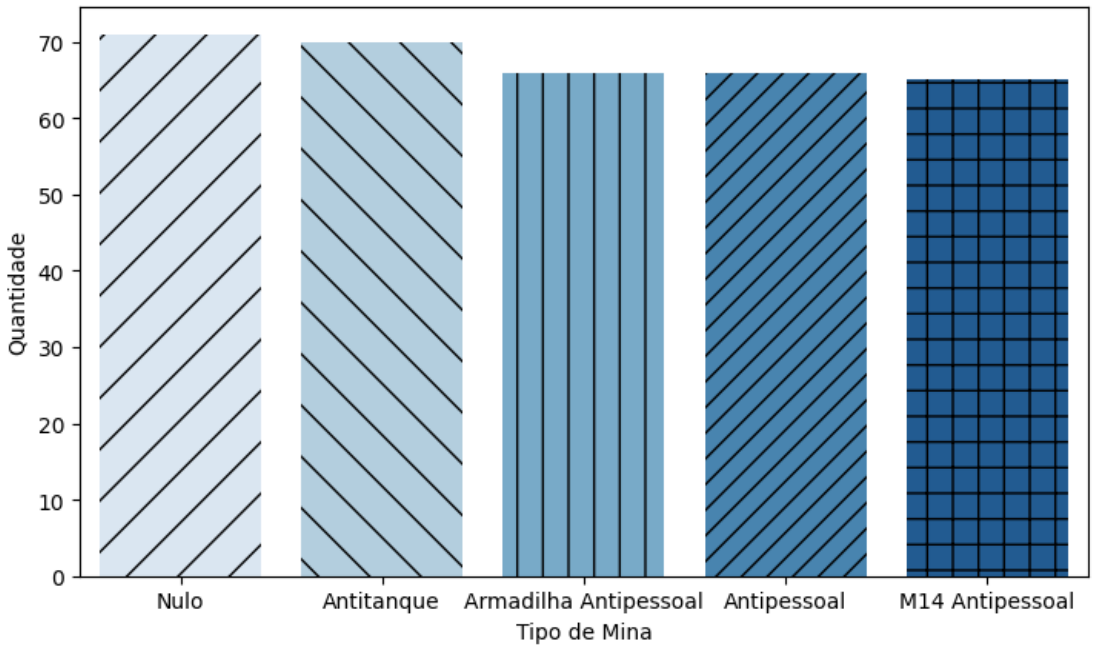
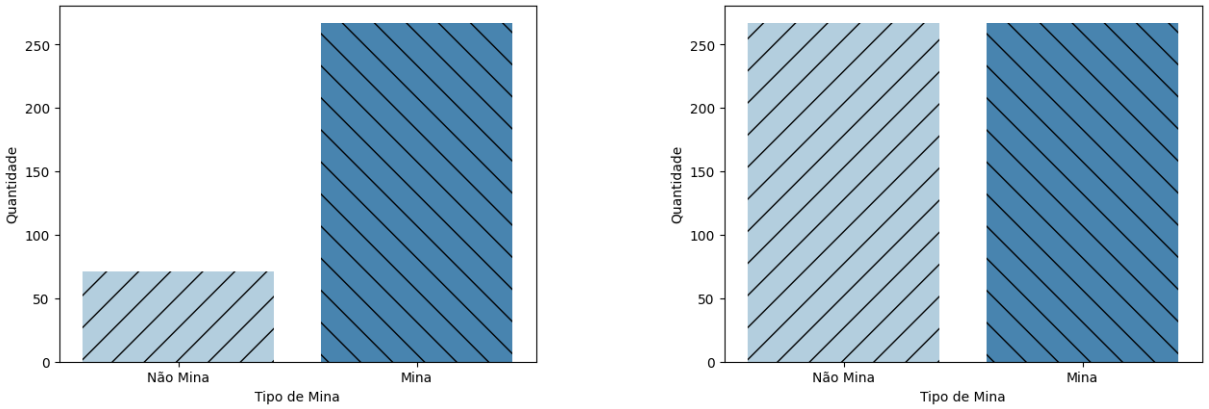


Figura 6 – Distribuição das classes do *Dataset*

O balanceamento do `Dataset` foi concretizado por meio do algoritmo *Synthetic Minority Over-sampling Technique* (SMOTE), o qual gera instâncias intermediárias entre dados semelhantes de forma sintética na classe minoritária (AZANK, 2022). De acordo com Fernandez et al. (2018), SMOTE é um algoritmo de *oversampling* com o objetivo de equilibrar conjuntos de treinamento desequilibrados. Em vez de simplesmente duplicar as instâncias da classe minoritária, o SMOTE gera novos exemplos sintéticos através da interpolação entre várias instâncias da classe minoritária que estão próximas umas das outras, levando em consideração os atributos. O objetivo é aumentar o número de exemplos da classe minoritária de forma a apresentar novas instâncias generalizadas. Após o rebalanceamento, o conjunto de dados consiste em 267 exemplos de cada classe, totalizando 534 instâncias, conforme mostrado na Figura 7b.



(a) Pós-transformação (b) Pós-balanceamento

Figura 7 – Distribuição das classes do *Dataset* pós pré-processamento

Na Tabela 4, são apresentadas amostras do `Dataset` após passar pelos processos de transformação e rebalanceamento. Ao término dessas etapas, o conjunto totalizou 534 instâncias, das quais 196 foram geradas sinteticamente. Note que, após a execução dos processos, a categorização foi simplificada, restando apenas as classes "Minas" e "Não Minas".

Tabela 4 – Amostra dos dados pós transformado e balanceado

Voltagem (*Voltage*)	**Altura (*Height*)**	**Tipo de Solo (*Soil Type*)**	**Tipo de Mina (*Mine Type*)**
0.338157	0.000000	1	0
0.335347	0.181818	4	0
0.283988	0.181818	2	0
0.310399	0.454545	3	0
0.295015	0.181818	6	0
0.999999	0.000000	1	1
0.504531	0.818182	4	1
0.438066	0.545455	2	1
0.999999	0.545455	3	1
0.303262	1.000000	1	1

Após todo o processo de transformação e balanceamento, chegamos a uma etapa crucial: a seleção de características. Nesta etapa, buscamos identificar quais atributos do conjunto de dados são mais relevantes para a tarefa de detecção de minas terrestres. A seleção dos melhores atributos se deu a partir de uma série de análises e testes. Começamos plotando um gráfico conhecido como *Pairplot*, o mesmo plota diversos *Scatterplots* (Gráficos de dispersão) para cada uma das combinações de duas características presentes no `Dataset`. Segundo Mayorga e Gleicher (2013), um *Scatterplot* é uma representação visual simples e intuitiva de dados bidimensionais de pontos. Ele mostra a relação entre duas variáveis, permitindo a identificação de tendências, correlações e *outliers*. Analisando os gráficos de dispersão, na Figura 8, podemos observar de maneira gráfica o relacionamento e a dispersão entre atributos. Além do *Pairplot*, também foram plotadas matrizes de correlação e variância dos atributos em busca de campos que apresentassem certa relação e dependência, ou seja,

atributos que são parecidos ou proporcionais uns com os outros. Observando a Figura 9a e Figura 9b, podemos notar que a maioria dos atributos não possui uma relação forte com os outros, mas a voltagem se destaca. Este atributo possui uma correlação de 50% com o tipo de mina, indicando que é um dos atributos mais importantes para a classificação.

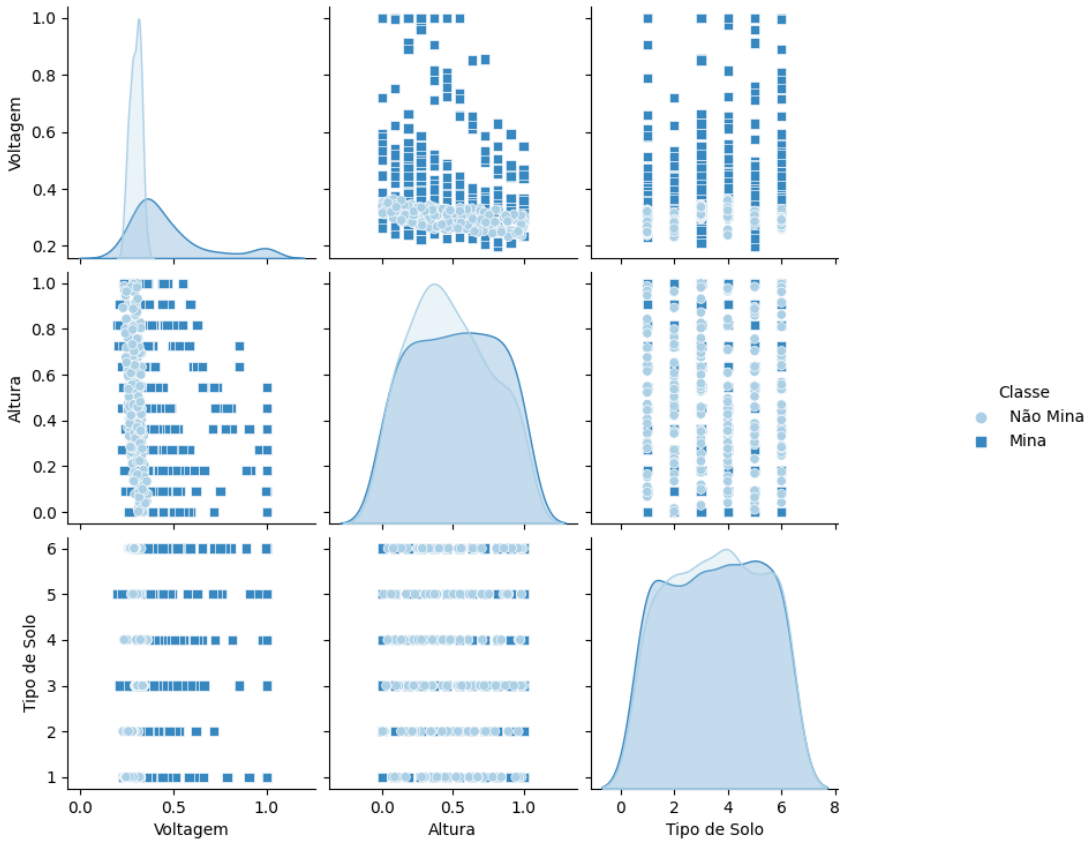
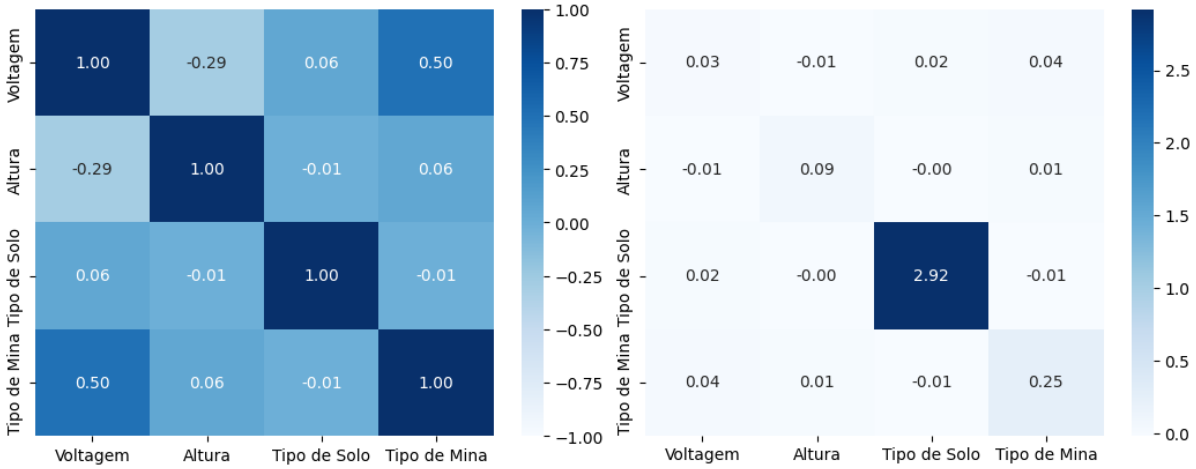


Figura 8 – Gráficos de dispersão dos atributos do `Dataset`



(a) Matriz de Correlação (b) Matriz de Covariância

Figura 9 – Relações entre atributos

Também foi utilizado um algoritmo de *Machine Learning* chamado *Extra Trees Classifier*, que nos permite visualizar em porcentagem a importância de uma característica,

ou seja, o quanto esse atributo é utilizado em cada árvore da floresta. Quando executado, o Código 2.1 apresenta os seguintes resultados, "Voltagem", "Altura" e "Tipo de solo" têm, respectivamente, 74%, 18% e 7% de importância para o modelo de árvore de decisão. Como o `Dataset` apresenta três atributos e somente a voltagem não era suficiente para

Código 2.1 – Extração da importância das características

```
X = df[["Voltage", "Height", "Soil Type"]].to_numpy()
y = df["Mine Type"]

model = ExtraTreesClassifier()
model.fit(X,y)

print(model.feature_importances_)
```

uma boa classificação, foram realizados testes para comparar a precisão de modelos de *Machine Learning* com diferentes combinações de características, e a combinação mais promissora foi aquela que utilizou todos os atributos.

2.4 Seleção de Algoritmos e Modelos

A seleção dos algoritmos e modelos de *Machine Learning* foi realizada após uma análise detalhada das características do problema e do conjunto de dados de detecção de minas terrestres, bem como dos requisitos do sistema. Levando em consideração o tamanho e a correlação dos dados no conjunto de dados, foi escolhido um conjunto de algoritmos de classificação com base em experimentos preliminares que avaliaram o desempenho de cada modelo em um conjunto de validação. Os algoritmos escolhidos foram:

- `Árvore de Decisão`, conforme descrito por Mahesh (2020), é uma representação gráfica de escolhas e suas consequências, organizada em uma estrutura de árvore. Os nós na árvore representam eventos ou escolhas, enquanto as arestas representam regras ou condições de decisão. Cada árvore é composta por nós e ramos, em que cada nó representa atributos pertencentes a um grupo a ser classificado, e cada ramo representa um valor que o nó pode assumir. Este modelo foi escolhido para o treinamento devido à sua capacidade de interpretabilidade, habilidade para lidar com dados complexos, versatilidade, além do fato de o `Dataset` apresentar apenas três atributos, resultando em árvores menores e mais simples.

- *K-nearest Neighbors* (KNN), KNN é um algoritmo simples de *Machine Learning* supervisionado utilizado para resolver problemas de classificação e regressão. É fácil de implementar e entender, mas apresenta uma grande desvantagem: sua significativa lentidão à medida que o tamanho dos dados aumenta (MAHESH, 2020). O algoritmo

organiza as instâncias em um plano cartesiano, utilizando os valores dos atributos escolhidos como coordenadas X e Y. Ao inserir uma instância para classificação, o algoritmo analisa a distância desta instância em relação às instâncias mais próximas. A instância pertencerá à classe da qual a maioria de seus "vizinhos" faz parte. O KNN foi escolhido para o treinamento deste trabalho devido à sua simplicidade e capacidade de identificar e explorar as relações entre dados.

- `Random Forest`, segundo Cutler, Cutler e Stevens (2012), `Random Forest` é um algoritmo que utiliza árvores de decisão como elementos básicos para fazer classificações. Cada árvore na floresta representa um "voto" sobre como classificar os resultados. Essas árvores são criadas a partir da escolha de amostras aleatórias dos dados. Ao final, as árvores são combinadas, e uma votação é realizada, sendo a classe com o maior número de votos selecionada como resultado final. Esse método resulta em uma classificação mais precisa devido à aleatoriedade na seleção dos dados utilizados por cada árvore. Alguns dos critérios considerados para a escolha do `Random Forest` incluem sua capacidade de reduzir a probabilidade de *overfitting* ao conjunto de dados, evitando viés no resultado final, além de fornecer diversas perspectivas diferentes.

2.5 Treinamento do Modelo

Para que o treinamento fosse realizado de forma a obter os melhores resultados, duas técnicas eram de suma importância. A primeira técnica é chamada de `Validação Cruzada K-fold`. Segundo Berrar (2019), A `Validação Cruzada K-fold` é um método em que o conjunto de treinamento é dividido em k subconjuntos distintos, chamados *Folds*, sem sobreposição. O modelo é treinado em $(k-1)$ *Folds* e testado no *Fold* restante. Esse processo é repetido (k) vezes com conjuntos de treinamento e teste diferentes, com o objetivo de avaliar a precisão de modelos de *Machine Learning* com diferentes conjuntos de dados. A segunda técnica é chamada de busca em grade (*Grid Search*).

Código 2.2 – Conjunto de algoritmos e seus respectivos hiperparâmetros para o treinamento

```
8  model = {
9      "KNN": (KNeighborsClassifier(), {"n_neighbors": [3,5,7]}),
10     "Árvore de decisão": (DecisionTreeClassifier(), {"max_depth": [3,5,7], "
       criterion": ["gini", "entropy"]}),
11     "RandomForest": (RandomForestClassifier(), {"n_estimators": [100,500,1000], "
       max_depth": [3,5,7], "criterion": ["gini", "entropy"]})
12 }
```

A busca em grade é o método otimização de hiperparâmetros, realizando uma exploração completa de um subconjunto do espaço de hiperparâmetros do algoritmo de treinamento

procurando a combinação com a maior precisão (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019). O conjunto escolhido de hiperparâmetros para este treinamento pode ser observado no Código 2.2.

Código 2.3 – Treinamento dos modelos

```
for model_name, (clf, parameters) in model.items():
    means = {'accuracy': [], 'recall': []}

    for fold, (train,test) in enumerate(kf.split(X,y)):
        best = GridSearchCV(clf, parameters, n_jobs=-1, cv=kf, scoring="recall",
    return_train_score=True)
        best.fit(X[train], y[train])

        y_pred = best.predict(X[test])

        means['accuracy'].append(accuracy_score(y[test], y_pred))
        means['recall'].append(recall_score(y[test], y_pred))

        print(f"{fold+1} -- {model_name} -- \
        acc: {round(means['accuracy'][-1] * 100, 2)}% rec: {round(means['recall
    '][-1] * 100, 2)}% Best parameters {best.best_params_}")

    print(f"\n\tA média de acuracia do {model_name} foi de {sum(means['accuracy'])/
    len(means['accuracy'])*100}%")
    print(f"\tA média de recall do {model_name} foi de {sum(means['recall'])/len(
    means['recall'])*100}%")
```

O processo de treinamento ocorre percorrendo uma variável que contém todos os algoritmos e seus respectivos hiperparâmetros. Nesse processo, as técnicas de busca em grade e validação cruzada são usadas simultaneamente, dividindo o `Dataset` em partes de treino e teste estratificadas, as quais garantem uma representação equitativa das classes de mina terrestre e não mina em cada conjunto. Conforme mostrado no Código 2.3, todo o processo anterior é feito ao mesmo tempo em que a classe `Grid Search CV` do `Scikit-learn` busca os melhores parâmetros, utilizando iterações cuidadosas para otimizar o desempenho e evitar o *overfitting* do modelo com base nos conjuntos de treino e teste.

2.6 Avaliação de Desempenho

Para avaliar o desempenho dos modelos treinados anteriormente, foi utilizado da Matriz de Confusão, além de algumas métricas relevantes para o problema de detecção de minas terrestres. Segundo Lago (2021), a Matriz de Confusão é uma ferramenta essencial na avaliação de algoritmos de classificação, a mesma é gerada a partir dos resultados obtidos por um modelo de *Machine Learning* e organiza as previsões em quatro categorias:

Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos. Essas categorias representam os acertos e erros do modelo em relação aos casos positivos e negativos de uma amostra. Esta abordagem permite não apenas a quantificação das métricas, como Acurácia e Revocação (*Recall*), mas também proporciona *insights* visuais sobre o desempenho do modelo. Um exemplo desta matriz esta presente na Figura 10[5].

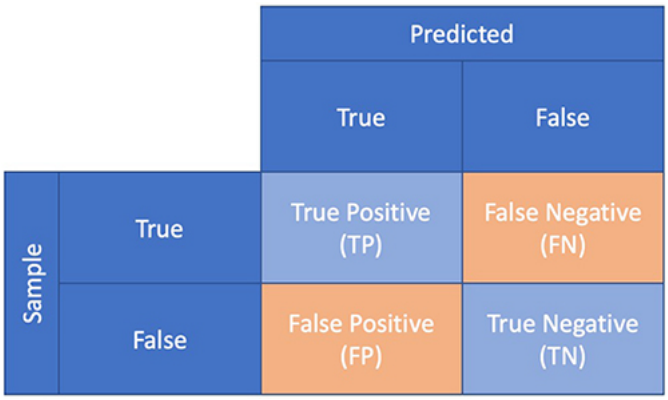


Figura 10 – Exemplo de uma Matriz de Confusão

A primeira métrica de desempenho foi a Acurácia. De acordo com Carvalho, Pereira e Cardoso (2019), a Acurácia está ligada à precisão das previsões quando se trata de dados que o modelo não conheceu anteriormente. A mesma calcula a taxa de acertos com base nos Verdadeiros Positivos e Verdadeiros Negativos, sua fórmula matemática pode ser observada na Equação 2.1.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.1}$$

No entanto, a métrica mais importante utilizada no trabalho foi o *Recall*. A utilização dessa métrica é fundamental para o problema em questão, uma vez que enfatiza a importância de minimizar os Falsos Negativos, como mostrado na Equação 2.2. Imagine que a aplicação identifique a ausência de minas terrestres em determinado solo, quando, na realidade, elas estão presentes, isso caracterizaria um Falso Negativo e poderia acarretar em diversos problemas, incluindo riscos à vida de quem estivesse analisando o solo. Portanto, é de suma importância utilizar a métrica de *Recall* como parâmetro durante o treinamento.

$$Revocação = \frac{VP}{VP + FN} \tag{2.2}$$

Avaliando os resultados contidos nas Figura 11a, Figura 11b, Figura 11c e Figura 12, podemos notar que um modelo se destaca em relação aos demais: o `Random Forest`. Esse obteve métricas de Acurácia de 93.4% e *Recall* de 89.5%, levando a um menor número de Falsos Negativos em comparação com os outros. Devido ao seu melhor desempenho

[5] <https://medium.com/@bernardolago/matriz-de-confusÃčo-7c0e36468323>

em relação aos modelos de `Árvore de Decisão` e KNN, o `Random Forest` foi escolhido como o algoritmo de *Machine Learning* para o treinamento da Inteligência Artificial deste projeto.

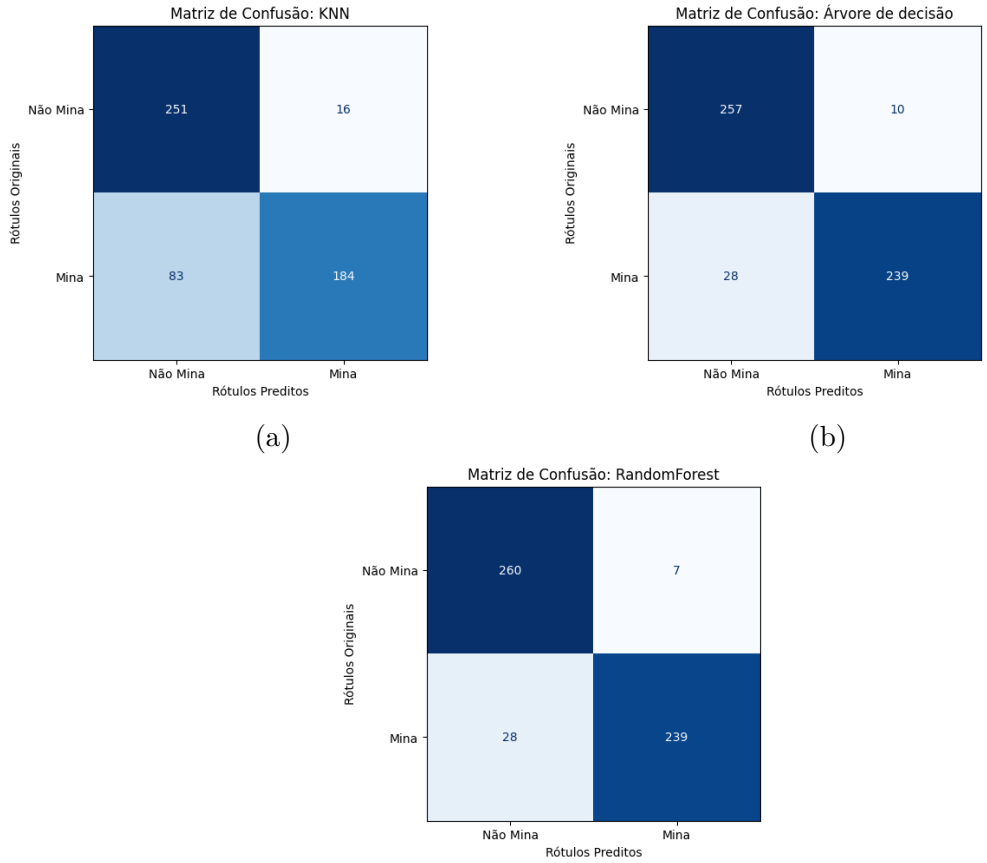


Figura 11 – Matrizes de Confusão dos Modelos

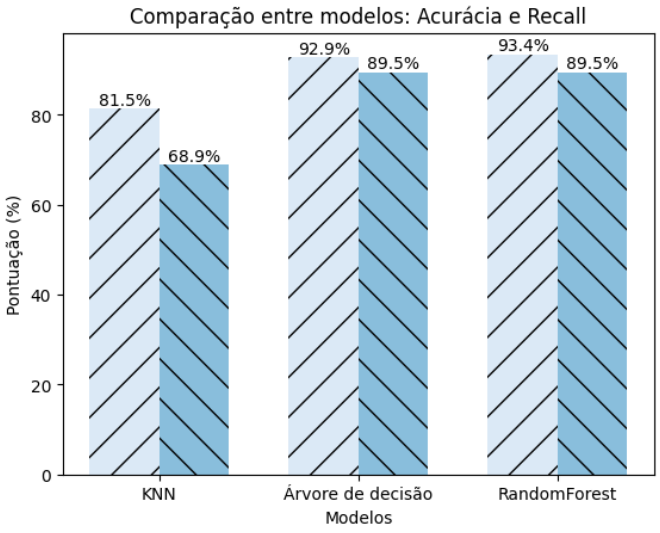


Figura 12 – Resultados de Acurácia e *Recall* dos modelos

2.7 Otimização e Ajuste de Hiperparâmetros

Para aprimorar ainda mais o desempenho do modelo (`Random Forest`), foram realizadas otimizações e ajustes de hiperparâmetros. Segundo AWS (2023), os hiperparâmetros são parâmetros externos para controlar o processo de treinamento de um modelo de *Machine Learning*. Os mesmos são frequentemente chamados de hiperparâmetros do modelo e devem ser configurados manualmente antes de iniciar o treinamento. Os hiperparâmetros podem influenciar diretamente o desempenho de um modelo, como sua precisão, a quantidade de camadas, algoritmo escolhido, entre outros. Ainda segundo AWS (2023), não existe um conjunto de regras bem definidas acerca de quais hiperparâmetros são a melhor escolha para um modelo, por isso é necessário realizar testes para encontrar o conjunto de parâmetros ideais.

Para encontrar as configurações que maximizam as métricas de desempenho do modelo `Random Forest`, foi criado um vetor contendo diversos hiperparâmetros e seus respectivos valores. Os principais hiperparâmetros escolhidos foram:

- `N-estimators`, para este hiperparâmetro foram selecionados os valores 50, 100, 150 e 500.
- `Max-depth`, com os seguintes valores *None*, 5, 7, 9, 11, 13 e 15.
- `Criterion`, para este hiperparâmetro foram escolhidos dois valores: “gini” e “entropy”.

Código 2.4 – Obtenção da melhor combinação de hiperparâmetros

```
30 params = {
31     'n_estimators' : [50, 100, 150, 500],
32     'max_depth'    : [None, 3, 5, 7, 9, 11, 13, 15],
33     'class_weight' : [None, 'balanced', 'balanced_subsample'],
34     'criterion'    : ['gini', 'entropy'],
35     'max_features' : [None, 'sqrt', 'log2'],
36     'min_samples_split': [2, 5, 10],
37     'min_samples_leaf': [1, 2, 4],
38     'oob_score': [False, True],
39 }
40
41 best = GridSearchCV(RandomForestClassifier(), params, n_jobs=-1, cv=kf, scoring="
       recall", return_train_score=True)
42 best.fit(X, y)
43
44 print("\n O Melhor estimador:\n", best.best_estimator_)
45 print("\n O melhor score:\n", best.best_score_)
46 print("\n Os melhores parametros:\n", best.best_params_)
```

Após a definição dos possíveis hiperparâmetros para o treinamento do modelo, torna-se necessário testar cada um desses parâmetros com seus respectivos valores para determinar quais são as melhores combinações a fim de obter o melhor desempenho possível. Nesta etapa do processo, como é observado no Código 2.4, é necessária a utilização da busca em grade, que percorre uma lista com todos os hiperparâmetros, e a utilização do *Recall* como métrica de pontuação para encontrar a combinação de valores que melhor se adequa ao modelo. A métrica escolhida visa maximizar o desempenho e diminuir a ocorrência de Falsos Negativos do modelo. No caso da `Grid Search CV`, os seguintes hiperparâmetros e valores foram encontrados: `criterion` "entropy", `max-depth` 15, `max-features` "log2", `min-samples-split` 5, `n-estimators` 50, `oob-score` *True*.

3 RESULTADOS

Antes de prosseguir para a coleta dos resultados conclusivos deste trabalho, uma série de passos foi seguida. Primeiramente, realizou-se um novo treinamento com o modelo de *Machine Learning* `Random Forest`, utilizando do conjunto de hiperparâmetros otimizados encontrados na seção 2.7 para maximizar os resultados do modelo. Neste treinamento,

Código 3.1 – Obtenção dos resultados do modelo de `Random Forest`

```
47 fig, ax = plt.subplots(figsize=(10, 10))
48 ax.plot([0, 1], [0, 1], "k--", label="chance level (AUC = 0.5)")
49
50 means = {'accuracy': [], 'recall': []}
51 confusion_matrices = []
52
53 clf = RandomForestClassifier(criterion='entropy', max_depth=15, max_features='log2'
       , min_samples_split=5, n_estimators=50, oob_score=True)
54
55 for fold, (train,test) in enumerate(kf.split(X,y)):
56     clf.fit(X[train], y[train])
57     y_pred = clf.predict(X[test])
58
59     accuracy = accuracy_score(y[test], y_pred)
60     recall = recall_score(y[test], y_pred)
61     means['accuracy'].append(accuracy)
62     means['recall'].append(recall)
63
64     confusion_matrices.append(confusion_matrix(y[test], y_pred))
65
66     viz = RocCurveDisplay.from_estimator(clf, X[test], y[test], name=f"ROC fold {
       fold + 1}", alpha=0.3, lw=1, ax=ax)
67
68     fig2, ax2 = plt.subplots(figsize=(10, 5))
69     ConfusionMatrixDisplay.from_predictions(y[test], y_pred, ax=ax2)
70     ax2.set_title(f"Confusion Matrix - Fold {fold + 1}")
71
72 aggregate_confusion_matrix = np.sum(confusion_matrices, axis=0)
73 fig3, ax3 = plt.subplots(figsize=(10, 5))
74 ConfusionMatrixDisplay(aggregate_confusion_matrix, display_labels=clf.classes_).
       plot(ax=ax3)
75 ax3.set_title("Aggregate Confusion Matrix")
76
77 print(f"\nAccuracy Score\t {sum(means['accuracy']) / len(means['accuracy']) * 100}"
       )
78 print(f"Recall Score\t {sum(means['recall']) / len(means['recall']) * 100}")
```

foi utilizada a `Validação Cruzada K-fold` para separar os conjuntos de treino e teste em cinco *Folds* a fim de obter e comparar as métricas de cada *Fold*. A avaliação dos resultados do treinamento será focada diretamente na quantidade de Falsos Negativos que o modelo apresentará.

Consequentemente, avaliou-se a métrica de *Recall*, já que a mesma nos permite realizar a análise da ocorrência de Falsos Negativos indicados pelo modelo retreinado. Segundo Foresti (2023), esta técnica mede o número de instâncias positivas que foram corretamente identificadas pelo modelo em relação ao número total de instâncias positivas no Dataset. Para auxiliar nesta avaliação, foram gerados gráficos como a curva ROC. A Curva ROC destaca a relação entre a taxa de Verdadeiros Positivos (Sensibilidade) e a taxa de Falsos Positivos (Especificidade), oferecendo uma visão ampla da capacidade do modelo em distinguir entre as classes. A precisão do modelo aumenta quando a especificidade e sensibilidade se aproximam de 1 (GÖNEN et al., 2006). O último gráfico utilizado para auxiliar na análise foi a matriz de confusão, esta matriz apresenta a quantidade total de Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos que os modelos alcançaram durante os *Fold*.

Tabela 5 – Métricas de Acurácia e *Recall* obtidas no treinamento

Fold	**Acurácia**	***Recall***
1	96.26168224299066	98.11320754716981
2	93.45794392523365	88.67924528301887
3	96.26168224299066	94.44444444444444
4	90.65420560747664	83.33333333333334
5	93.39622641509435	92.45283018867924

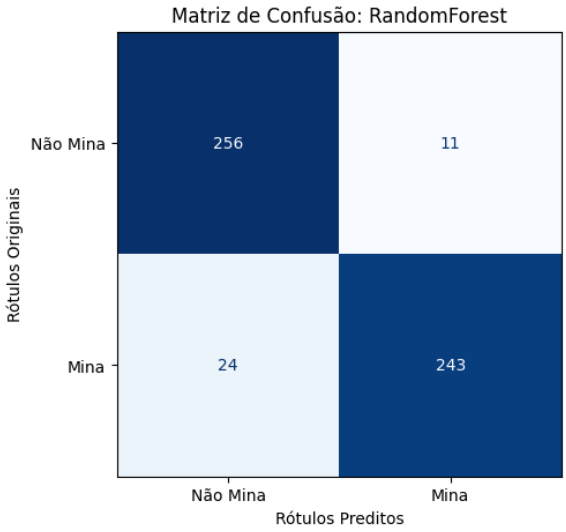


Figura 13 – Matriz de Confusão

Com todas as métricas decididas e a metodologia de análise de resultados definida, podemos realizar uma análise quantitativa do desempenho do modelo de `Random Forest`. O código utilizado para o treinamento e obtenção de resultados do modelo pode ser

observado no Código 3.1. O treinamento do modelo resultou em uma média de 94,00% de Acurácia, com uma variação de 90% a 96%, e uma porcentagem de *Recall* de 91,40%, com variação entre 88% a 98%, conforme apresentado na Tabela 5. Realizando uma comparação com o desempenho do modelo sem a otimização de hiperparâmetros, tivemos um ganho de 0,6% de acurácia e 1,9% de *Recall*, o que comprova que o processo de otimização feito anteriormente foi efetivo e apresentou resultados satisfatórios.

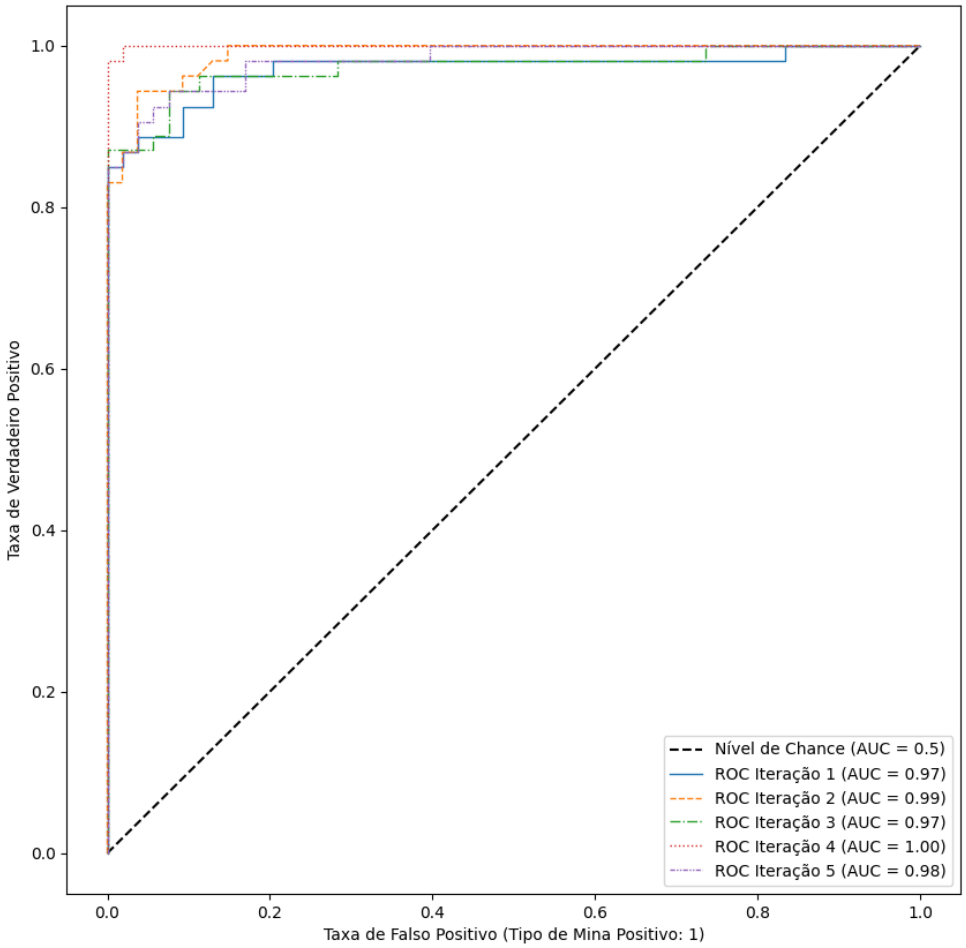


Figura 14 – Curva ROC

Ao comparar este resultado com os trabalhos de Pryshchenko et al. (2022), Núñez-Nieto et al. (2014), nos quais ambos propuseram a utilização de Redes Neurais, observamos que obtiveram métricas de Acurácia 4% e 2% menor, respectivamente, em relação ao presente trabalho. Já o trabalho de Pradana (2023), que utiliza o mesmo `Dataset`, obteve uma Acurácia 10,18% menor. Além da Acurácia, o *Recall* obteve resultados altos, mas que causam certo receio, já que, segundo Foresti (2023), o *Recall* aumenta à medida que há menos Falsos Negativos. Obter 91% desta métrica implica que o modelo está raramente apontando que não há nada em lugares onde existem minas terrestres. Podemos visualizar a quantidade de Falsos Negativos na Figura 13. Nesta matriz de confusão, é possível identificar que, das 534 classificações, 11 foram Falsos Positivos e 24 foram Falsos Negativos, representando, respectivamente, 2,05% das instâncias classificadas como

Falsos Positivos e 4,49% como Falsos Negativos. Fazendo um comparativo com o trabalho de Makki et al. (2018), observa-se que com a utilização de Redes Neurais, os mesmos conseguiram zerar o número de Falsos Positivos, o que indica que há espaço para melhorias. Analisando a curva ROC na Figura 14, podemos notar que a probabilidade de classificação de Verdadeiros Positivos do modelo está alta, o que, para este trabalho, são resultados muito satisfatórios. O notebook contendo todo o código desenvolvido para este trabalho está disponível no Github[1].

[1] <https://github.com/Bielgomes/Deteccao-de-Minas-Terrestres-uma-abordagem-com-inteligencia-artificial>

4 CONCLUSÕES

Este trabalho teve como objetivo principal treinar um modelo de *Machine Learning* capaz de identificar a presença ou ausência de minas terrestres em áreas contaminadas. A abordagem adotada envolveu o uso de técnicas de *Machine Learning*. A escolha do modelo `Random Forest`, combinada com a `Validação Cruzada K-fold` e o emprego de um `Dataset` montado a partir de dados obtidos por sensores FLC, permitiu um treinamento eficaz, conforme foi demonstrado no Capítulo 3.

Os resultados obtidos no decorrer deste estudo demonstram que o modelo final apresentou métricas animadoras, com uma média de 94,00%, que é até 4% maior comparados com os trabalhos Pryshchenko et al. (2022), Núñez-Nieto et al. (2014), de Acurácia e um *Recall* médio de 91,40%. A análise da curva ROC revela que o modelo apresenta uma boa capacidade de distinguir entre as classes. A matriz de confusão demonstra que, dos 534 exemplos classificados, onde 4,49% eram Falsos Negativos, representando uma baixa porcentagem.

Em conclusão, este trabalho contribuiu para a aplicação de técnicas de *Machine Learning* na detecção de minas terrestres, demonstrando resultados acima da média. Entretanto, é importante reconhecer que nenhum método isolado de detecção de minas atinge 100% de eficiência, sendo necessária a combinação de diversas técnicas para alcançar melhores resultados. Além disso, o conjunto de dados utilizado não é ideal para obter resultados mais precisos e deverá ser expandido posteriormente. Conclui-se que este trabalho contribui para os esforços humanitários na descontaminação de áreas afetadas por minas terrestres e demonstra o potencial que a Inteligência Artificial e *Machine Learning* possuem nesta área.

4.1 Trabalhos futuros

Como trabalho futuro, propomos realizar uma nova análise e treinamento de uma Inteligência Artificial utilizando uma versão expandida do `Dataset` deste estudo. A intenção é aprimorar a diversidade e representatividade de cada classe, buscando um possível aumento nas métricas. Além disso, exploraremos alternativas ao algoritmo `Random Forest` e de técnicas de pré-processamento de dados para melhorar o desempenho do modelo. A expansão do conjunto de dados buscará abranger diversos cenários relacionados à presença ou ausência de minas terrestres, contribuindo para um aprendizado mais amplo do modelo. Refazer o treinamento do modelo com outras técnicas de *Machine Learning* e um `Dataset` maior e mais abrangente poderia nos auxiliar a concretizar melhorias na

eficiência do modelo e, consequentemente, na sua aplicabilidade.

Uma possível extensão deste projeto envolveria desenvolvimento de um veículo ou ferramenta onde embarcaríamos a Inteligência Artificial desenvolvida neste trabalho. Essa plataforma seria equipado com sensores FLC para registrar distorções eletromagnéticas do solo. Além disso, o dispositivo seria capaz de selecionar o tipo de solo da área em análise e incluiria sensores infravermelhos para medir a distância em relação ao solo. Essa configuração possibilitaria a identificação da presença ou ausência de minas terrestres no solo em questão. A expansão do projeto para essa aplicação prática permitiria a utilização em operações de desminagem, fornecendo informações em tempo real sobre a presença de minas terrestres, contribuindo assim para uma abordagem mais eficiente e segura.

Outro possível trabalho seria a condução de testes simulados para avaliar o desempenho da Inteligência Artificial em ambientes controlados. Esses testes poderiam simular diversas condições de solo e presença de minas terrestres, proporcionando uma análise mais detalhada da capacidade do modelo em diferentes cenários. Além disso, essa abordagem forneceria *insights* valiosos sobre o comportamento do modelo em condições específicas, fornecendo uma maior compreensão das limitações e pontos fortes do modelo, e assim, possibilitando a identificação de possíveis melhorias.

REFERÊNCIAS

ACHKAR, R.; OWAYJAN, M.; MRAD, C. Landmine detection and classification using mlp. In: *2011 Third International Conference on Computational Intelligence, Modelling Simulation*. [S.l.: s.n.], 2011. p. 1–6. Citado na página 13.

ANDRAOS, L. V.; MARANHÃO, P. H. C.; GUEDES, R. O. de C. Revisão de técnicas de detecção de minas terrestres. *Revista Militar de Ciência e Tecnologia*, v. 35, n. 2, p. 42–48, 2018. Citado 2 vezes nas páginas 15 e 16.

AWS. *o que é o ajuste de hiperparâmetros? — métodos de ajuste de hiperparâmetros explicados*. 2023. Disponível em: <https://aws.amazon.com/pt/what-is/hyperparameter-tuning/>. Citado na página 33.

AZANK, F. *Dados Desbalanceados - O que são e como evitá-los*. Turing Talks, 2022. Disponível em: <https://medium.com/turing-talks/dados-desbalanceados-o-que-s%C3%A3o-e-como-evit%C3%A1-los-43df4f49732b>. Citado na página 25.

BERRAR, D. Cross-validation. In: RANGANATHAN, S. et al. (Ed.). *Encyclopedia of Bioinformatics and Computational Biology*. Oxford: Academic Press, 2019. p. 542–545. ISBN 978-0-12-811432-2. Disponível em: <https://www.sciencedirect.com/science/article/pii/B978012809633820349X>. Citado na página 29.

CARVALHO, D. V.; PEREIRA, E. M.; CARDOSO, J. S. Machine learning interpretability: A survey on methods and metrics. *Electronics*, v. 8, n. 8, 2019. Disponível em: <https://www.mdpi.com/2079-9292/8/8/832>. Citado na página 31.

CUTLER, A.; CUTLER, D. R.; STEVENS, J. R. Random forests. *Ensemble machine learning: Methods and applications*, Springer, p. 157–175, 2012. Citado na página 29.

FERNANDEZ, A. et al. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, v. 61, p. 4, 2018. Citado na página 25.

FORESTI, T. *Recall em machine learning: Conceitos E Aplicações*. Awari, 2023. Disponível em: <https://awari.com.br/recall-em-machine-learning-conceitos-e-aplicacoes/>. Citado 2 vezes nas páginas 36 e 37.

FOSTER, A. *As pessoas que arriscam suas vidas para Limpar Campos Minados*. BBC, 2022. Disponível em: <https://www.bbc.com/portuguese/internacional-64112812>. Citado na página 15.

GÖNEN, M. et al. Receiver operating characteristic (roc) curves. *SAS Users Group International (SUGI)*, Citeseer, v. 31, p. 210–231, 2006. Citado na página 36.

KRIGER, D. *O que É python, para que serve e por que aprender?* Kenzie Academy Brasil, 2023. Disponível em: <https://kenzie.com.br/blog/o-que-e-python/>. Citado na página 20.

LAGO, B. *Matriz de Confusão*. Medium, 2021. Disponível em: <https://medium.com/@bernardolago/matriz-de-confus%C3%A3o-7c0e36468323>. Citado na página 30.

LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019. Citado na página 30.

MACDONALD, J. et al. *Alternatives for landmine detection*. [S.l.]: Rand Santa Monica, CA, 2003. Citado na página 16.

MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, v. 9, n. 1, p. 381–386, 2020. Citado na página 28.

MAKKI, I. et al. Rbf neural network for landmine detection in h yperspectral imaging. In: *2018 7th European Workshop on Visual Information Processing (EUVIP)*. [S.l.: s.n.], 2018. p. 1–6. Citado 4 vezes nas páginas 13, 17, 18 e 38.

MARCELA. Awari, 2023. Disponível em: <https://awari.com.br/scikit-learn>. Citado na página 21.

MAYORGA, A.; GLEICHER, M. Splatterplots: Overcoming overdraw in scatter plots. *IEEE transactions on visualization and computer graphics*, IEEE, v. 19, n. 9, p. 1526–1538, 2013. Citado na página 26.

MIRANDA, J. V. d. *Jupyter Notebook: Exemplos de Códigos e Como Usar*. Alura, 2021. Disponível em: <https://www.alura.com.br/artigos/conhecendo-o-jupyter-notebook>. Citado 2 vezes nas páginas 21 e 22.

NUNES, W. V. Detecção de minas terrestres por radiação penetrante. *Universidade Federal do Rio de Janeiro*, 2005. Citado na página 16.

NúñEZ-NIETO, X. et al. Gpr signal characterization for automated landmine and uxo detection based on machine learning techniques. *Remote Sensing*, v. 6, n. 10, p. 9729–9748, 2014. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/6/10/9729>. Citado 4 vezes nas páginas 17, 18, 37 e 39.

ONU. *Novo relatório mostra aumento em número de vítimas de minas terrestres | ONU News*. United Nations, 2022. Disponível em: <https://news.un.org/pt/story/2022/11/1805742>. Citado 2 vezes nas páginas 13 e 14.

PRADANA, G. A. *Deteksi dan Klasifikasi Ranjau Darat*. Kaggle, 2023. Disponível em: <https://www.kaggle.com/code/gadipradana123/deteksi-dan-klasifikasi-ranjau-darat>. Citado 2 vezes nas páginas 13 e 37.

PRYSHCHENKO, O. A. et al. Implementation of an artificial intelligence approach to gpr systems for landmine detection. *Remote Sensing*, v. 14, n. 17, 2022. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/14/17/4421>. Citado 4 vezes nas páginas 13, 17, 37 e 39.

RAMYACHITRA, D.; MANIKANDAN, P. Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, v. 5, n. 4, 2014. Citado na página 25.

SUN, Y.; LI, J. Adaptive learning approach to landmine detection. *IEEE transactions on aerospace and electronic systems*, IEEE, v. 41, n. 3, p. 973–985, 2005. Citado 2 vezes nas páginas 17 e 18.

WATERHOUSE, J. *Guerra na Ucrânia: O drama das minas terrestres que deixam milhares de mortos e mutilados no conflito*. BBC, 2023. Disponível em: <https://www.bbc.com/portuguese/articles/c29renpxgjzo>. Citado na página 14.