

DataSynth: Documentação Técnica

Seu Nome

June 21, 2024

Abstract

DataSynth é uma ferramenta projetada para anonimizar dados financeiros sensíveis utilizando ruído branco gaussiano. Esta documentação técnica explica o funcionamento do DataSynth, incluindo a geração de ruído branco, a aplicação de anonimização e a recuperação de dados encriptados.

Contents

1	Introdução	1
2	Geração de Ruído Branco	1
2.1	Transformação de Box-Muller	2
2.2	Ajuste da Intensidade do Ruído	2
3	Aplicação do Ruído Branco	2
3.1	Exemplo de Código	2
4	Visualização do Ruído Branco	2
5	Recuperação de Dados	3
6	Considerações Finais	3

1 Introdução

DataSynth é uma solução inovadora para a anonimização de dados financeiros. Utilizando ruído branco gaussiano, o DataSynth preserva as propriedades estatísticas dos dados enquanto protege informações sensíveis.

2 Geração de Ruído Branco

A geração de ruído branco no DataSynth é realizada utilizando a transformação de Box-Muller. Esta transformação gera variáveis aleatórias gaussianas (normais) a partir de variáveis uniformemente distribuídas.

2.1 Transformação de Box-Muller

A transformação de Box-Muller é utilizada para converter duas variáveis aleatórias uniformemente distribuídas $U1$ e $U2$ em duas variáveis gaussianas $Z0$ e $Z1$ com média zero e desvio padrão 1.

$$Z0 = \sqrt{-2 \ln U1} \cos(2\pi U2) \quad (1)$$

$$Z1 = \sqrt{-2 \ln U1} \sin(2\pi U2) \quad (2)$$

Essas variáveis gaussianas são então escaladas para ajustar a intensidade do ruído.

2.2 Ajuste da Intensidade do Ruído

O ruído gerado é ajustado por um fator de escala definido pelo usuário. Este fator de escala afeta diretamente o desvio padrão do ruído aplicado, enquanto a média permanece zero.

$$ruído = Z0 \times ruído_nível \times std_original \quad (3)$$

3 Aplicação do Ruído Branco

O ruído branco gaussiano é aplicado aos dados financeiros para anonimização. O nível de ruído pode ser ajustado para controlar a intensidade do ruído aplicado.

3.1 Exemplo de Código

```
def adicionar_ruído_branco_e_encryptar(dados, coluna, chave, ruído_nível=1):
    tamanho = len(dados)
    Z0 = gerar_ruído_branco(tamanho)
    dados[coluna] = dados[coluna].apply(lambda x: float(str(x).replace(',', ' ', '.)))
    ruído = Z0 * ruído_nível * dados[coluna].std()
    dados_sinteticos = dados.copy()
    dados_sinteticos[coluna] += ruído

    # Encryptar os dados originais
    fernet = Fernet(chave)
    dados['Encrypted_' + coluna] = dados[coluna].apply(lambda x: fernet.encrypt(str(x).encode()))
    return dados_sinteticos, dados
```

4 Visualização do Ruído Branco

O DataSynth permite a visualização do ruído branco gerado e da autocorrelação do ruído. Essas visualizações ajudam a entender como o ruído é aplicado e como ele afeta os dados.

```
def plot_ruido_branco(ruido, ruido_nivel, std_original):
    media_ruido = np.mean(ruido)
    desvio_padrao_ruido = np.std(ruido)

    fig, ax = plt.subplots(figsize=(10, 4))
    sns.histplot(ruido, bins=30, kde=True, ax=ax)
    ax.axvline(media_ruido, color='r', linestyle='--', label=f'Média do Ruído: {media_ruido}')
    ax.axvline(desvio_padrao_ruido, color='g', linestyle='--', label=f'Desvio Padrão do Ruído: {desvio_padrao_ruido}')
    ax.set_xlabel('Valor')
    ax.set_ylabel('Frequência')
    ax.set_title(f'Ruído Branco Gaussiano Gerado (Nível de Ruído = {ruido_nivel})')
    ax.legend()
    st.pyplot(fig)
```

5 Recuperação de Dados

O DataSynth permite a recuperação dos dados originais utilizando uma chave de deciptação. Esta funcionalidade garante que os dados possam ser revertidos para sua forma original de maneira segura, se necessário.

```
def decriptar_dados(dados, chave, colunas_texto):
    fernet = Fernet(chave)
    for coluna in dados.columns:
        if coluna.startswith('Encrypted_'):
            original_coluna = coluna.replace('Encrypted_', '')
            if original_coluna in colunas_texto:
                try:
                    dados[original_coluna] = dados[coluna].apply(lambda x: fernet.decrypt(x))
                except Exception as e:
                    st.write(f"Erro ao decriptar a coluna {coluna}: {e}")
            else:
                try:
                    dados[original_coluna] = dados[coluna].apply(lambda x: float(fernet.decrypt(x)))
                except Exception as e:
                    st.write(f"Erro ao decriptar a coluna {coluna}: {e}")
    # Remover colunas encriptadas após deciptação
    colunas_encriptadas = [col for col in dados.columns if col.startswith('Encrypted_')]
    dados.drop(columns=colunas_encriptadas, inplace=True)
    return dados
```

6 Considerações Finais

DataSynth é uma ferramenta poderosa para anonimização de dados financeiros, garantindo privacidade e conformidade com as regulamentações. Com a visual-

ização do ruído branco e a capacidade de recuperação de dados, o DataSynth oferece uma solução completa e transparente.