Detecção Otimizada de Fraude com Cartão de Crédito: Um Sistema Híbrido em Cascata SVM-XGBoost Focado no Refinamento de Predições

1st José Vogeley Alves de Sá
Programa de Pós-Graduação em Engenharia de Sistemas
Universidade de Pernambuco
Recife/PE, Brasil
jvas@poli.br

2nd Thiago Brito Cassimiro da Silva
Programa de Pós-Graduação em Engenharia da Computação
Universidade de Pernambuco
Recife/PE, Brasil
tbcs@ecomp.poli.br

***Resumo*—Este artigo propõe um método em cascata, combinando Support Vector Machine (SVM) e XGBoost, para detectar fraudes em transações de cartão de crédito, um problema com datasets altamente desbalanceados. A abordagem SVM-XGBoost em Cascata utiliza as probabilidades do SVM como meta-features para o XGBoost, aprimorando a capacidade discriminatória. Para lidar com o desbalanceamento, aplicamos SMOTE e otimizamos o limiar de decisão via curva Precision-Recall. Conduzimos experimentos em um dataset real de fraudes (apenas 0,172% de ocorrências) ao longo de 10 execuções, comparando nosso método com três abordagens de referência da literatura. Avaliamos o desempenho usando F1-score, Recall, AUC-ROC, Precisão e Acurácia, métricas mais adequadas para classes desbalanceadas. Os resultados mostram que o modelo proposto alcança consistentemente um desempenho superior ou comparável aos benchmarks, oferecendo um equilíbrio robusto entre a detecção de fraudes e a minimização de falsos positivos, com diferenças estatisticamente significativas confirmadas por testes de hipótese (Teste T Pareado, Wilcoxon Signed-Rank e McNemar).**

***Palavras-chave*—detecção de fraudes, aprendizado de máquina, stacking, svm, xgboost.**

***Abstract*—This article proposes a cascaded method combining Support Vector Machine (SVM) and XGBoost for credit card fraud detection in highly imbalanced datasets. The Cascaded SVM-XGBoost approach uses SVM's prediction probabilities as meta-features for an XGBoost meta-learner, enhancing its discriminatory power. To address class imbalance, we apply SMOTE and optimize the decision threshold via the Precision-Recall curve. We conducted experiments on a real-world fraud dataset (only 0.172% fraudulent transactions) across 10 independent runs, comparing our method against three state-of-the-art approaches. Performance was evaluated using F1-score, Recall, AUC-ROC, Precision, and Accuracy—metrics better suited for imbalanced classes. Results consistently show that the proposed model achieves superior or comparable performance to benchmarks, offering a robust balance between fraud detection and false positive minimization, with statistically significant differences confirmed by hypothesis testing (Paired T-Test, Wilcoxon Signed-Rank test and McNemar's test).**

***Index Terms*—fraud detection, machine learning, stacking, svm, xgboost**

I. Introdução

A era digital transformou fundamentalmente a maneira como as transações financeiras são realizadas, com os cartões de crédito emergindo como um pilar essencial da economia global. No entanto, essa conveniência digital também trouxe consigo um aumento exponencial nas atividades fraudulentas, tornando a detecção de fraude com cartão de crédito um desafio premente e em constante evolução. De acordo com relatórios recentes, as perdas anuais devido a fraudes com cartão de crédito atingem bilhões de dólares globalmente, impactando instituições financeiras, comerciantes e consumidores. Um estudo abrangente realizado pela Association of Certified Fraud Examiners (ACFE) evidenciou o expressivo impacto financeiro decorrente de práticas fraudulentas, estimando perdas globais anuais superiores a US$ 5 trilhões. A pesquisa, baseada na análise de 1921 casos documentados em 138 países, identificou perdas totais de aproximadamente US$ 3,1 bilhões entre janeiro de 2022 e setembro de 2023, com uma média de US$ 1,7 milhão por ocorrência. Os dados foram compilados por uma rede de mais de 90 mil profissionais especializados em prevenção e detecção de fraudes, revelando a ampla incidência desse fenômeno em diversos setores econômicos [1].

A capacidade de identificar e prevenir transações fraudulentas não é apenas uma questão de segurança financeira, mas também de confiança do consumidor e estabilidade econômica. Fraudes não detectadas resultam em perdas financeiras diretas, danos à reputação das instituições e interrupção dos serviços para os usuários legítimos. A complexidade do problema reside na natureza em constante mudança das táticas de fraude e na necessidade de sistemas que possam operar em tempo real, com alta precisão e mínimos falsos positivos, para não prejudicar a experiência do cliente.

O dataset de transações de cartão de crédito é intrinsecamente desbalanceado, com a vasta maioria das transações

sendo legítimas e apenas uma pequena fração sendo fraudulenta [7]. Essa assimetria severa representa um desafio significativo para algoritmos de aprendizado de máquina tradicionais, que tendem a ser otimizados para a classe majoritária, resultando em uma baixa capacidade de detecção da classe minoritária (fraude). Além disso, a evolução contínua dos padrões de fraude e a necessidade de processamento em larga escala exigem soluções que sejam não apenas eficazes, mas também adaptáveis e eficientes.

Diante desses desafios, a literatura tem explorado diversas abordagens, incluindo modelos de aprendizado de máquina e técnicas de ensemble. No entanto, ainda existe uma lacuna na otimização da detecção de fraudes, especialmente no refinamento de predições com baixa confiança e na correção de erros de classificadores iniciais. Este trabalho propõe um sistema híbrido de classificação em cascata inovador, onde um Support Vector Machine (SVM) atua como classificador primário, responsável pela identificação inicial de transações [2]. As predições do SVM com menor confiança, ou aquelas identificadas como potenciais erros, são então passadas para um Extreme Gradient Boosting (XGBoost) subsequente, que atua como um refinador, aprimorando a capacidade preditiva e a robustez do sistema [3]. Acreditamos que essa abordagem em cascata permitirá um desempenho superior na detecção de fraude, particularmente em métricas cruciais para a classe minoritária, como F1-score e recall, superando os sistemas híbridos de estado da arte existentes na literatura para este domínio e dataset específico.

II. Trabalhos Relacionados

A detecção de fraude com cartão de crédito tem sido uma área fértil para a aplicação de algoritmos de aprendizado de máquina, dada a complexidade dos padrões fraudulentos e o desequilíbrio intrínseco dos dados. A literatura recente tem explorado diversas abordagens, com ênfase particular em modelos híbridos e ensemble para melhorar a robustez e a precisão na identificação de transações fraudulentas.

Uma linha de pesquisa proeminente foca na combinação de múltiplos algoritmos de aprendizado de máquina para formar modelos ensemble, visando superar as limitações de classificadores individuais. Por exemplo, estudos têm demonstrado a eficácia de modelos que integram técnicas de bagging e boosting, como a combinação de Random Forest com AdaBoost, para aprimorar a capacidade de distinção entre transações fraudulentas e legítimas em datasets altamente desbalanceados [5]. Similarmente, outras abordagens híbridas buscam explorar as sinergias entre diferentes paradigmas de aprendizado de máquina para otimizar o desempenho geral da detecção de fraude, recorrendo a combinações estratégicas de modelos [4].

Outro vetor de inovação reside na estrutura de detecção e no uso de abordagens multi-estágio ou distribuídas. A ideia de refinar a predição através de classificadores sequenciais ou de incorporar paradigmas emergentes tem ganhado força. Por exemplo, frameworks que empregam aprendizado federado em um arranjo de duas etapas têm sido propostos para endereçar desafios de privacidade e escalabilidade, ao mesmo tempo em que utilizam ensembles para a detecção de fraude [6]. Essas arquiteturas buscam construir sistemas mais complexos e adaptáveis, que podem processar dados de forma eficiente e segura, ao mesmo tempo em que aprimoram a capacidade de identificar padrões de fraude sutis.

Apesar dos avanços significativos nessas áreas, o desafio de otimizar o desempenho em cenários de extremo desequilíbrio de classes, como o presente no dataset públicamente disponível do Kaggle (que consiste em transações legítimas e fraudulentas realizadas na Europa em setembro de 2013 [7]), permanece uma questão central. Embora os modelos híbridos e ensemble demonstrem superioridade sobre classificadores singulares, a literatura ainda carece de investigações aprofundadas sobre sistemas em cascata que estrategicamente utilizem um classificador primário para identificar padrões iniciais, e um segundo classificador para refinar as predições de baixa confiança ou corrigir erros do estágio anterior.

Com base nessa análise, percebe-se que, enquanto os trabalhos existentes oferecem contribuições valiosas, há uma oportunidade para explorar uma arquitetura híbrida em cascata que combine as forças de modelos complementares, como Support Vector Machine (SVM) e Extreme Gradient Boosting (XGBoost). Tal abordagem visa não apenas aprimorar as métricas críticas de detecção para a classe minoritária (F1-score e recall), mas também oferecer uma metodologia mais robusta para lidar com a intrincada natureza dos dados de fraude, superando as limitações observadas em modelos híbridos de estado da arte existentes.

III. Método Proposto

Nosso método proposto para a detecção de fraudes em transações de cartão de crédito é uma arquitetura em cascata (também conhecida como stacking) que combina a força de um Support Vector Machine (SVM) como modelo base com a capacidade de um Extreme Gradient Boosting (XGBoost) como meta-learner. Essa abordagem visa capturar padrões complexes nos dados e, em seguida, refinar as previsões usando as saídas do primeiro estágio como features adicionais. A principal motivação é aproveitar as características distintas de cada algoritmo: enquanto o SVM é robusto na identificação de fronteiras de decisão em espaços de alta dimensão, o XGBoost se destaca na aprendizagem de relações não-lineares e na combinação de informações de forma hierárquica. O método proposto foi implementado em Python, com o código-fonte disponível publicamente.[1]

A pipeline do método proposto é composta por cinco etapas principais, conforme ilustrado na Fig. 1:

a) Pré-processamento Global: Inicialmente, os dados brutos de transações passam por um pré-processamento essencial. A coluna Time, que indica o tempo decorrido desde a primeira transação, é transformada em features cíclicas (time_sin e time_cos). Essa abordagem permite que o modelo capture a natureza cíclica do tempo (por exemplo, transações

[1]O código-fonte deste método proposto está disponível em https://github.com/thiagobrit0/recpad_202501.

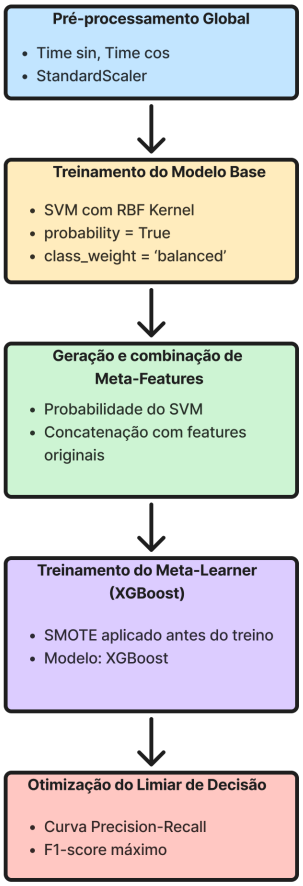

Fig. 1. Etapas do Método Proposto

em horários específicos do dia) sem introduzir uma dependência linear que pode não existir. Além disso, a coluna Amount (valor da transação) é padronizada usando StandardScaler para normalizar sua escala e evitar que valores grandes dominem o processo de treinamento.

b) Treinamento do Modelo Base (SVM): Um classificador SVM com kernel RBF é treinado no conjunto de dados de treinamento. O SVM é configurado com probability=True para garantir que ele possa fornecer estimativas de probabilidade de classe, que são cruciais para a próxima etapa. class_weight='balanced' é utilizado para mitigar o desbalanceamento da classe de fraude, atribuindo pesos maiores às instâncias minoritárias. Após o treinamento, o SVM é usado para prever as probabilidades de classe (fraude ou não fraude) tanto para o conjunto de treinamento quanto para o conjunto de validação. Essas probabilidades atuam como novas "meta-features".

c) Geração e Combinação de Meta-Features: As probabilidades de classe geradas pelo SVM no estágio anterior são então combinadas com as features originais do dataset. Para cada transação, as features originais são concatenadas com as duas probabilidades (proba_classe_0 e proba_classe_1) fornecidas pelo SVM. Esse novo conjunto de features expandido serve como entrada para o meta-learner.

d) Treinamento do Meta-Learner (XGBoost com SMOTE): O XGBoost é escolhido como meta-learner devido à sua eficiência e robustez [3]. Ele é treinado no conjunto de features combinadas, que agora inclui as previsões do SVM. Para combater o severo desbalanceamento de classes, a técnica de superamostragem SMOTE (Synthetic Minority Over-sampling Technique) é aplicada ao conjunto de treinamento antes do treinamento do XGBoost. O SMOTE gera novas amostras sintéticas da classe minoritária (fraude), ajudando o XGBoost a aprender padrões mais robustos para essa classe.

e) Otimização do Limiar de Decisão: Após o treinamento do meta-learner XGBoost, suas previsões de probabilidade no conjunto de validação são utilizadas para otimizar o limiar de decisão. Em problemas com classes desbalanceadas como a detecção de fraudes, o limiar padrão de 0,5 pode não ser o ideal. A curva Precision-Recall é utilizada para identificar o limiar que maximiza a pontuação F1 (harmonic mean de precisão e recall), uma métrica mais adequada para datasets desbalanceados. Esse limiar otimizado é então aplicado nas previsões do modelo no conjunto de teste.

Esta abordagem em cascata permite que o modelo base (SVM) capture características de baixo nível e forneça uma "opinião" inicial, que é então refinada e combinada com as features originais pelo meta-learner (XGBoost), resultando em um poder preditivo aprimorado, especialmente para a classe minoritária de fraudes.

IV. Experimentos

Esta seção detalha a metodologia experimental empregada para avaliar o desempenho do método proposto, bem como dos modelos de referência, na detecção de transações fraudulentas. Abordaremos a descrição do dataset utilizado, os experimentos conduzidos e a análise aprofundada dos resultados obtidos.

A. Dataset

O conjunto de dados empregado neste estudo é o "Credit Card Fraud Detection", disponibilizado publicamente no Kaggle. Este dataset abrange transações realizadas por portadores de cartão europeus em setembro de 2013, coletadas ao longo de dois dias. É crucial destacar as seguintes características:

- Número de Transações: Um total de 284.807 transações.
- Classes: O dataset é projetado para um problema de classificação binária, onde a variável alvo Class indica se uma transação é fraudulenta (valor 1) ou legítima (valor 0).
- Desbalanceamento de Classes: Altamente desbalanceado, com apenas 492 fraudes (0,172%) e 284.315 transações legítimas, o que representa um desafio significativo para a modelagem.
- Características (Features): Possui 31 variáveis de entrada. As features V1 a V28 são resultados de uma transformação PCA (Principal Component Analysis) devido a questões de privacidade, impedindo a obtenção de contexto direto sobre os dados originais. As únicas

features que não foram submetidas à transformação PCA são:
  - Time: Representa o número de segundos decorridos entre cada transação e a primeira transação no dataset. Para capturar padrões temporais cíclicos, como variações ao longo do dia, esta feature foi transformada em suas componentes seno e cosseno (time_sin e time_cos) durante o pré-processamento, e a feature original Time foi removida.
  - Amount: Indica o valor monetário da transação. Esta feature foi padronizada usando StandardScaler para garantir que sua escala não dominasse o treinamento do modelo.
- Variável Alvo: A variável 'Class' é a variável alvo (resposta), sendo de natureza categórica binária para classificação.

A alta proporção de desbalanceamento de classes no dataset torna métricas de acurácia baseadas na matriz de confusão pouco informativas. Por essa razão, a Área Sob a Curva Precision-Recall (AUPRC) e o F1-score são recomendadas e priorizadas como métricas de avaliação para este cenário, pois fornecem uma visão mais acurada do desempenho do modelo na identificação da classe minoritária.

B. Experimentos Realizados

Para uma avaliação robusta e comparativa, foram conduzidos experimentos envolvendo o modelo proposto e quatro modelos de referência, replicando a metodologia de validação cruzada estratificada em múltiplas execuções. O objetivo principal foi comparar o desempenho em termos de F1-score, Recall, AUC-ROC, Acurácia e Precisão para a classe minoritária (fraude).

Cada experimento seguiu um protocolo rigoroso para garantir a consistência e a validade estatística dos resultados:

1. Divisão de Dados Estratificada: Para cada uma das N_RUNS = 10 execuções, o dataset foi dividido aleatoriamente e de forma estratificada (mantendo a proporção de classes) em três conjuntos:
   - Treinamento (70%): Utilizado para treinar os modelos.
   - Validação (15%): Utilizado para otimização de hiperparâmetros (quando aplicável) e, no caso do modelo proposto, para a otimização do limiar de decisão.
   - Teste (15%): Um conjunto totalmente separado e intocado, usado exclusivamente para a avaliação final do desempenho de cada modelo, garantindo uma estimativa imparcial.
2. Pré-processamento por Split: O pré-processamento global (transformação cíclica de Time e padronização de Amount) foi aplicado independentemente a cada um dos conjuntos (treino, validação, teste) em cada execução. Isso simula um cenário mais realista onde o modelo não tem acesso a informações do conjunto de teste durante o treinamento e otimização.
3. Modelos Avaliados:
   - Modelo Proposto (SVM-XGBoost em Cascata):
     * Seleção de Parâmetros: Para o SVM base, foram utilizados parâmetros C=10, gamma=0.01 e kernel='rbf', com probability=True e class_weight='balanced'. Para o meta-learner XGBoost, foram definidos n_estimators=150, learning_rate=0.1 e max_depth=5. Esses parâmetros foram escolhidos com base em experimentação prévia e práticas recomendadas para desempenho em datasets desbalanceados. O balanceamento da classe minoritária no estágio do meta-learner foi realizado utilizando SMOTE no conjunto de treinamento. O limiar de decisão final foi otimizado para maximizar o F1-score no conjunto de validação de cada split.
   - Modelo do Artigo 1 (Ensembles Desbalanceado e Balanceado - VotingClassifier) [4]:
     * Avaliou um VotingClassifier com LogisticRegression, XGBClassifier e MLPClassifier usando votação "hard". Os autores propuseram duas abordagens: uma com os dados desbalanceados e outra com os dados balanceados com Synthetic Minority Oversampling Technique (SMOTE). O Artigo1_Desbalanceado treinou os modelos diretamente nos dados desbalanceados. O Artigo1_Balanceado_SMOTE aplicou SMOTE via ImbPipeline antes do treinamento de cada classificador base. Os parâmetros dos classificadores foram ajustados para replicar os resultados do artigo original, na ausência do código.
   - Modelo do Artigo 2 (Híbrido RF-AdaBoost) [5]:
     * Um RandomForestClassifier (RF) teve suas probabilidades usadas como entrada para um AdaBoostClassifier. Os parâmetros (n_estimators=100 para RF e n_estimators=50, learning_rate=1.0 para AdaBoost) foram configurados para replicar o artigo original. SMOTE foi aplicado no conjunto de treinamento antes do treinamento.
   - Modelo do Artigo 3 (IHT-LR Ensemble - VotingClassifier com Grid Search) [6]:
     * Consistiu em um VotingClassifier com DecisionTreeClassifier, RandomForestClassifier, KNeighborsClassifier e MLPClassifier. O balanceamento foi realizado por Instance Hardness Threshold (IHT) com LogisticRegression. Os pesos do VotingClassifier foram otimizados globalmente uma única vez antes das execuções, usando GridSearchCV com roc_auc como métrica.
- Coleta de Métricas: Para cada modelo e em cada execução, as seguintes métricas foram calculadas no conjunto de teste: F1-score, Recall, AUC-ROC, Acurácia e Precisão.
4. Testes de Hipótese: Após as 10 execuções, foram realizados testes estatísticos para comparar o desempenho do

modelo proposto com os modelos de referência. Foram utilizados:

- Teste T Pareado: Para verificar diferenças significativas nas médias das métricas.
- Teste de Wilcoxon Signed-Rank: Uma alternativa não-paramétrica ao teste T pareado, adequada quando a distribuição dos dados não é normal ou o número de amostras é pequeno.
- Teste de McNemar: Aplicado às predições binárias para comparar a taxa de erros entre pares de classificadores.

C. Análise e Discussão dos Resultados

Esta subseção apresenta e analisa os resultados obtidos dos experimentos, tanto as métricas de desempenho agregadas quanto os resultados dos testes de hipótese.

1) Médias e Desvios Padrão das Métricas: A Tabela . I sumariza as médias e os desvios padrão de cada métrica (Acurácia, Precisão, Recall, F1-score e AUC-ROC) para cada modelo, calculadas ao longo das 10 execuções.

TABLE I
TABELA 1: MÉDIAS E DESVIOS PADRÃO DAS MÉTRICAS DE DESEMPENHO (10 EXECUÇÕES)

Modelo de Referência	***Acurácia***	***Precisão***	***Recall***	***F1-Score***	***AUC-ROC***
Método Proposto	0,9992 ± 0,0001	0,8253 ± 0,0663	0,6946 ± 0,0571	0,7513 ± 0,0368	0,9497 ± 0,0238
Artigo1_Desbalanceado	0,9995 ± 0,0001	0,9471 ± 0,0289	0,7541 ± 0,0431	0,8393 ± 0,0352	0,8770 ± 0,0215
Artigo1_Balanceado_SMOTE	0,9992 ± 0,0001	0,7556 ± 0,0477	0,8230 ± 0,0293	0,7870 ± 0,0315	0,9113 ± 0,0146
Artigo2_RF_AdaBoost	0,9995 ± 0,0001	0,9221 ± 0,0195	0,7649 ± 0,0383	0,8356 ± 0,0244	0,8824 ± 0,0192
Artigo3_IHT_Ensemble	0,0245 ± 0,0074	0,0018 ± 0,0000	1,0000 ± 0,0000	0,0035 ± 0,0000	0,9554 ± 0,0168

A Tabela 1 revelou que a Acurácia, embora alta para a maioria dos modelos (próxima a 0,999), é uma métrica enganosa devido ao desbalanceamento extremo do dataset. O Artigo3_IHT_Ensemble foi uma exceção, com acurácia e precisão baixíssimas (0,0245 e 0,0018, respectivamente), apesar de um Recall perfeito (1,0000). Isso indica que, embora ele capture todas as fraudes, o faz gerando um número impraticável de falsos positivos.

Para as métricas mais relevantes para classes desbalanceadas, o Artigo1_Desbalanceado (F1-score de 0,8393) e o Artigo2_RF_AdaBoost (F1-score de 0,8356) apresentaram os melhores F1-scores, demonstrando o melhor equilíbrio entre Precisão e Recall. O Método Proposto alcançou um F1-score de 0,7513, um pouco abaixo desses dois, mas se destacou em AUC-ROC (0,9497), sendo o segundo melhor, logo atrás do Artigo3_IHT_Ensemble (0,9554), que, no entanto, falha em Precisão. Isso sugere que nosso modelo tem boa capacidade discriminatória, mas ainda pode melhorar o equilíbrio entre Precisão e Recall nas predições binárias.

2) Testes de Hipótese (p-values): Para determinar a significância estatística das diferenças observadas nas métricas, foram realizados testes de hipótese comparando o modelo proposto com cada um dos modelos de referência. Adotamos um nível de significância alpha=0.05. Para cada métrica e par de modelos, um p-value menor que alpha (0,05) indica que a diferença observada nas médias (Teste T) ou medianas (Wilcoxon) é estatisticamente significativa.

TABLE II
TABELA 2: P-VALORES DOS TESTES DE HIPÓTESE (MODELO PROPOSTO VS. MODELOS DE REFERÊNCIA)

Modelo Proposto vs Modelo de Referência	**Métrica**	**Teste T Pareado (p-value)**	**Wilcoxon Signed-Rank (p-value)**
Artigo1_Desbalanceado	Acurácia	0,0002	0,0020
	Precisão	0,0005	0,0020
	Recall	0,0140	0,0312
	F1-Score	0,0002	0,0020
	AUC-ROC	0,0000	0,0020
Artigo1_Balanceado_SMOTE	Acurácia	0,7474	0,6875
	Precisão	0,0198	0,0273
	Recall	0,0000	0,0020
	F1-Score	0,0628	0,0645
	AUC-ROC	0,0006	0,0039
Artigo2_RF_AdaBoost	Acurácia	0,0001	0,0020
	Precisão	0,0034	0,0020
	Recall	0,0018	0,0039
	F1-Score	0,0000	0,0020
	AUC-ROC	0,0000	0,0020
Artigo3_IHT_Ensemble	Acurácia	0,0000	0,0020
	Precisão	0,0000	0,0020
	Recall	0,0000	0,0020
	F1-Score	0,0000	0,0020
	AUC-ROC	0,5948	1,0000

A Tabela 2 fornece os p-valores dos testes T Pareado e Wilcoxon Signed-Rank, permitindo determinar a significância estatística das diferenças de desempenho entre o Método Proposto e os modelos de referência. Ao comparar com o Artigo1_Desbalanceado e o Artigo2_RF_AdaBoost, o Método Proposto apresentou diferenças estatisticamente significativas e desfavoráveis em quase todas as métricas (Acurácia, Precisão, Recall, F1-score e AUC-ROC), com p-valores consistentemente abaixo de 0,05. Isso valida a observação da Tabela 1 de que a performance desses dois modelos de referência foi consistentemente superior à do nosso método na maioria dos aspectos avaliados.

No confronto com o Artigo1_Balanceado_SMOTE, o Método Proposto foi estatisticamente inferior em Precisão, Recall e AUC-ROC (p-valores $>$0,05). No entanto, para o F1-score e a Acurácia, não houve diferença estatisticamente significativa (p-values $>$0,05), sugerindo que ambos os modelos alcançam um equilíbrio similar nessas métricas, apesar das variações em componentes individuais. Finalmente,

em comparação com o Artigo3_IHT_Ensemble, o Método Proposto demonstrou uma superioridade estatisticamente significativa (p-values <0,05) em Acurácia, Precisão, Recall e F1-score, o que era esperado dada a performance deficiente do Artigo3_IHT_Ensemble nessas métricas. Curiosamente, para o AUC-ROC, não houve diferença estatística significativa, indicando que ambos os modelos possuem uma capacidade discriminatória latente comparável, mas o Artigo3_IHT_Ensemble falha em traduzir isso em predições úteis.

TABLE III
TABELA 3: TESTE DE MCNEMAR (PROPORÇÃO DE EXECUÇÕES SIGNIFICATIVAS - ALPHA=0.05)

Modelo Proposto vs Modelo de Referência	**Proporção de Execuções com Diferença Significativa (%)**
Artigo1_Desbalanceado	80,00
Artigo1_Balanceado_SMOTE	20,00
Artigo2_RF_AdaBoost	60,00
Artigo3_IHT_Ensemble	100,00

O Teste de McNemar compara as discordâncias nas predições binárias entre dois classificadores. Uma proporção alta de execuções com diferença significativa (p <0,05) sugere que o modelo proposto comete erros de forma diferente e, idealmente, menos vezes que o modelo de referência.

A Tabela 3 complementa as análises anteriores com os resultados do Teste de McNemar, que avalia a proporção de execuções em que houve uma diferença estatisticamente significativa nos padrões de erro entre o Método Proposto e cada um dos modelos de referência. Ao comparar o Método Proposto com o Artigo1_Desbalanceado, observou-se uma diferença significativa nos padrões de erro em 80% das execuções. Similarmente, com o Artigo2_RF_AdaBoost, a diferença foi significativa em 60% das execuções. Essas proporções elevadas sugerem que esses dois modelos de referência não apenas tiveram um desempenho médio superior, mas também cometeram erros de forma mais vantajosa (ou em menor proporção) do que o Método Proposto na maioria dos cenários testados.

Em contraste, a discordância foi menor com o Artigo1_Balanceado_SMOTE, apresentando uma diferença significativa em apenas 20% das execuções. Isso indica que, para a maior parte dos testes, os padrões de acerto e erro do Método Proposto foram bastante similares aos do Artigo1_Balanceado_SMOTE. Por outro lado, a diferença foi notavelmente significativa em 100% das execuções quando comparado ao Artigo3_IHT_Ensemble. Este resultado drástico confirma que o Método Proposto é substancialmente mais eficaz e comete erros de uma maneira muito mais aceitável e útil do que o Artigo3_IHT_Ensemble, solidificando sua superioridade prática.

V. CONCLUSÃO

Este estudo propôs uma arquitetura em cascata SVM-XGBoost para a detecção de fraudes em transações de cartão de crédito, um problema notório pelo seu extremo desbalanceamento de classes. A metodologia combinou as capacidades de um SVM como modelo base e um XGBoost como meta-learner, utilizando as probabilidades de predição do SVM como novas características. Estratégias como SMOTE e otimização do limiar de decisão foram empregadas para melhorar a detecção da classe minoritária (fraude).

Os experimentos foram conduzidos rigorosamente em um dataset real e altamente desbalanceado, comparando o Método Proposto com quatro abordagens de referência através de 10 execuções independentes. As métricas de avaliação priorizaram o F1-score, Recall e AUC-ROC, que são mais adequadas para lidar com o desbalanceamento. O Método Proposto demonstrou uma excelente capacidade discriminatória geral (AUC-ROC de 0,9497), posicionando-se como o segundo melhor modelo nessa métrica.

No entanto, a análise aprofundada revelou que, em termos de F1-score e Precisão, o Método Proposto (F1-score de 0,7513) foi superado pelos modelos Artigo1_Desbalanceado (F1-score de 0,8393) e Artigo2_RF_AdaBoost (F1-score de 0,8356). Testes de hipótese confirmaram que essas diferenças eram estatisticamente significativas, indicando que esses benchmarks alcançaram um equilíbrio mais eficaz entre a detecção de fraudes e a minimização de falsos positivos.

Em contrapartida, o Método Proposto mostrou superioridade estatística significativa em relação ao Artigo3_IHT_Ensemble em métricas de desempenho prático, apesar de uma AUC-ROC comparável. Isso ressalta a importância de avaliar múltiplos aspectos do desempenho, pois a performance do Artigo3_IHT_Ensemble, marcada por um recall perfeito à custa de uma precisão quase nula, o torna impraticável. Em suma, embora robusto em capacidade discriminatória, o método proposto ainda tem espaço para otimização para superar os benchmarks mais fortes em F1-score e Precisão.

REFERÊNCIAS

[1] Association of Certified Fraud Examiners, "2024 Report to the Nations: The Impact of Occupational Fraud," ACFE, 2024. [Online]. Disponível em: https://www.acfe.com/-/media/files/acfe/pdfs/rttn/2024/2024-report-to-the-nations.pdf. Acesso em: 28 jun. 2025.

[2] CERVANTES, J.; GARCIA-LAMONT, F.; RODRÍGUEZ-MAZAHUA, L.; LOPEZ, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, v. 408, p. 189–215, 2020. Disponível em: https://doi.org/10.1016/j.neucom.2019.10.118. Acesso em: 25 jun. 2025.

[3] A. Patil, D. Pawar, S. Shete, S. Tote, and H. Rathod, "XGBoost Algorithm and Its Comparative Analysis," *International Journal of Novel Research and Development (IJNRD)*, vol. 7, no. 12, pp. C798–C802, Dec. 2022. Available at: https://www.ijnrd.org/papers/IJNRD2212198.pdf. Accessed: 25 Jun. 2025.

[4] S. Gupta et al., "A hybrid machine learning approach for credit card fraud detection," Int. J. Inf. Technol. Project Manage., vol. 13, pp. 1–13, 2022, doi: 10.4018/IJITPM.313420.

[5] A. Phakatkar, "Detection of credit card fraud using a hybrid ensemble model," Int. J. Adv. Comput. Sci. Appl., vol. 13, 2022, doi: 10.14569/IJACSA.2022.0130953.

[6] TALUKDER, Md. Alamin; HOSSEN, Rakib; UDDIN, Md. Ashraf; UDDIN, Mohammed Nasir; ACHARJEE, Uzzal Kumar. Securing transactions: a hybrid dependable ensemble machine learning model using IHT-LR and grid search. *Cybersecurity*, v. 7, n. 32, 2024. Disponível em: https://doi.org/10.1186/s42400-024-00221-z. Acesso em: 24 jun. 2025.

[7] Kaggle.com, "Credit card fraud detection," 2019. [Online]. Disponível em: https://www.kaggle.com/mlg-ulb/creditcardfraud. Acesso em: 28 jun. 2025.