Detecção de URLs Maliciosas Usando Machine Learning Clássico e Quântico

"Detection of malicious URLs using machine learning"

Nuria Reyes-Dorta Pino Caballero-Gil Carlos Rosa-Remedios

Wireless Networks (2024) Volume 30: 7543-7560

Universidade de La Laguna, Tenerife, Espanha

Olá a todos. Hoje, vou apresentar uma visão geral do artigo "Detecção de URLs maliciosas usando machine learning", publicado na revista Wireless Networks. Este trabalho explora e compara técnicas de aprendizado de máquina, tanto clássicas quanto quânticas, para combater um dos vetores de ataque cibernético mais comuns: o phishing.

Introdução: O Problema das URLs Maliciosas

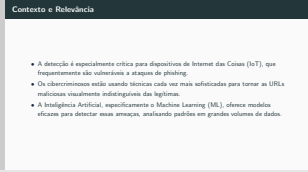
- URLs (Uniform Resource Locator) são os endereços únicos de cada página na Internet.
- Um dos ciberataques mais comuns utiliza versões fraudulentas de URLs para enganar os usuários.
- Esses links parecem legítimos, mas redirecionam para páginas falsas projetadas para roubar informações pessoais como senhas e dados bancários.
- O phishing se tornou o vetor de ataque inicial mais comum, de acordo com o relatório anual da Agência da União Europeia para a Cibersegurança (ENISA).

Introdução: O Problema das URLs Maliciosas
- URLs (Uniform Resource Locator) são os endereços únicos de cada página na Internet.
- Um dos ciberataques mais comuns utiliza versões fraudulentas de URLs para enganar os usuários.
- Esses links parecem legítimos, mas redirecionam para páginas falsas projetadas para roubar informações pessoais como senhas e dados bancários.
- O phishing se tornou o vetor de ataque inicial mais comum, de acordo com o relatório anual da Agência da União Europeia para a Cibersegurança (ENISA).

Para começar, vamos definir o problema. Todos nós usamos URLs todos os dias para navegar na web. Cibercriminosos exploram nossa confiança criando URLs fraudulentas que imitam sites legítimos. O objetivo é nos enganar para que forneçamos informações sensíveis. Essa técnica, conhecida como phishing, é extremamente prevalente e foi identificada pela ENISA como a ameaça inicial mais comum em ciberataques.

Contexto e Relevância

- A detecção é especialmente crítica para dispositivos de Internet das Coisas (IoT), que frequentemente são vulneráveis a ataques de phishing.
- Os cibercriminosos estão usando técnicas cada vez mais sofisticadas para tornar as URLs maliciosas visualmente indistinguíveis das legítimas.
- A Inteligência Artificial, especificamente o Machine Learning (ML), oferece modelos eficazes para detectar essas ameaças, analisando padrões em grandes volumes de dados.



O problema é ainda mais grave no contexto da Internet das Coisas, ou IoT. Muitos desses dispositivos estão conectados à internet, mas não possuem mecanismos de segurança robustos, tornando-os alvos fáceis. Com os criminosos aprimorando suas táticas, a detecção manual ou baseada em assinaturas simples não é mais suficiente. É aqui que o Machine Learning entra como uma ferramenta poderosa, capaz de aprender e identificar os padrões complexos que caracterizam uma URL maliciosa.

Objetivos da Pesquisa

- Analisar a aplicação de diferentes técnicas de ML para a detecção precoce de URLs fraudulentas.
- Prestar atenção especial à aplicação pioneira de Quantum Machine Learning (QML) para resolver este problema.
- Comparar os resultados obtidos com QML com aqueles produzidos por métodos clássicos de Machine Learning e Deep Learning.
- Avaliar o potencial do QML como uma nova ferramenta no campo da cibersegurança.

Objetivos da Pesquisa
- Analisar a aplicação de diferentes técnicas de ML para a detecção precoce de URLs fraudulentas.
- Prestar atenção especial à aplicação pioneira de Quantum Machine Learning (QML) para resolver este problema.
- Comparar os resultados obtidos com QML com aqueles produzidos por métodos clássicos de Machine Learning e Deep Learning.
- Avaliar o potencial do QML como uma nova ferramenta no campo da cibersegurança.

Os objetivos desta pesquisa são claros. O primeiro é fazer uma análise abrangente das técnicas de Machine Learning tradicionais. O segundo, e o ponto mais inovador deste artigo, é explorar o campo emergente do Quantum Machine Learning, ou QML. O terceiro objetivo é realizar uma comparação direta de desempenho entre as abordagens clássica e quântica. Finalmente, o trabalho busca avaliar se o QML, apesar de ser uma tecnologia recente, já pode oferecer vantagens práticas para a cibersegurança.

Metodologia: Uma Abordagem Iterativa

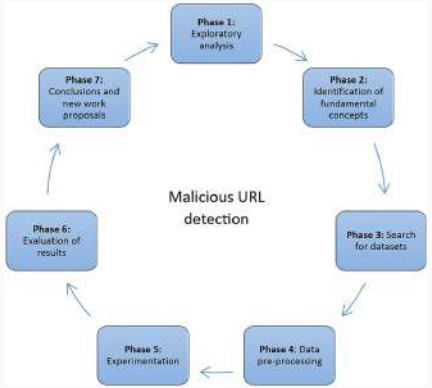
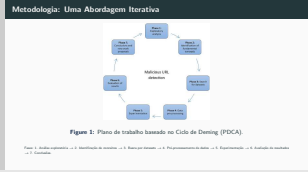


Figure 1: Plano de trabalho baseado no Ciclo de Deming (PDCA).

Fases: 1. Análise exploratória → 2. Identificação de conceitos → 3. Busca por datasets → 4. Pré-processamento de dados → 5. Experimentação → 6. Avaliação de resultados → 7. Conclusões.



Para atingir esses objetivos, os autores seguiram uma metodologia estruturada em sete fases, baseada no ciclo PDCA, ou Plan-Do-Check-Act. Este ciclo foi aplicado iterativamente, primeiro para o Machine Learning clássico e depois para o Quântico. As fases incluíram desde a análise exploratória e busca por datasets atualizados, passando pelo complexo pré-processamento e experimentação, até a avaliação final dos resultados e conclusões.

ML Clássico: Dados e Pré-processamento

- **Dataset:** Utilizou-se um dataset público e rotulado do repositório de machine learning da Universidade de Trieste.
- **Desafio:** O dataset era desbalanceado, com apenas 8.618 URLs fraudulentas em um total de 181.916 amostras. Foi necessário usar `class_weight="balanced"` nos modelos.
- **Engenharia de Features:**
  - Inicialmente, foram usadas features como `serverType`, `contentType`, etc.
  - Para melhorar a precisão, novas variáveis foram criadas a partir da própria URL, como: comprimento da URL, contagem de caracteres ('/', '.', '-', etc.).

ML Clássico: Dados e Pré-processamento
- Dataset: Utilizou-se um dataset público e rotulado do repositório de machine learning da Universidade de Trieste.
- Desafio: O dataset era desbalanceado, com apenas 8.618 URLs fraudulentas em um total de 181.916 amostras. Foi necessário usar class_weight="balanced" nos modelos.
- Engenharia de Features:
  - Inicialmente, foram usadas features como serverType, contentType, etc.
  - Para melhorar a precisão, novas variáveis foram criadas a partir da própria URL, como: comprimento da URL, contagem de caracteres ('/', '.', '-', etc.).

Na abordagem clássica, o ponto de partida foi um dataset público, o que é ótimo para a reprodutibilidade. No entanto, ele apresentava um desafio comum: o desbalanceamento de classes, com muito mais URLs benignas do que maliciosas. Isso foi tratado ajustando o peso das classes nos algoritmos. O passo crucial foi a engenharia de features. Uma análise inicial usando apenas os metadados do site não foi suficiente. A precisão melhorou significativamente após a criação de novas features extraídas diretamente da string da URL, como seu comprimento e a frequência de caracteres especiais.

ML Clássico: Melhoria com Engenharia de Features

Antes (Acurácia: 87.77%)

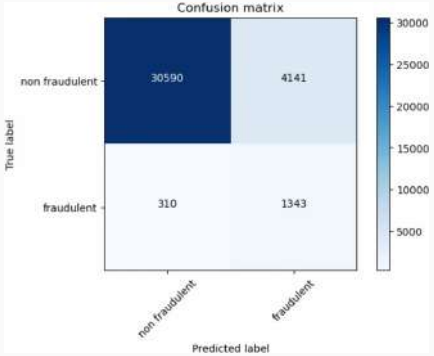


Figure 2: Matriz de confusão inicial.

Depois (Acurácia: 93.18%)

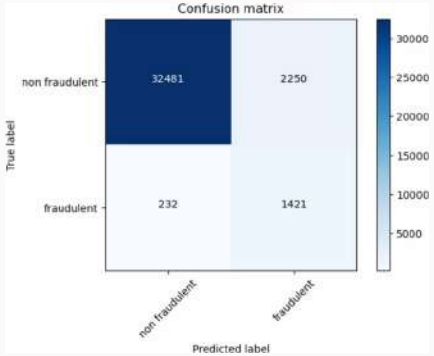
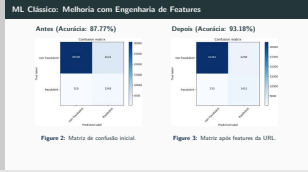


Figure 3: Matriz após features da URL.



Aqui podemos ver o impacto da engenharia de features visualmente. À esquerda, a matriz de confusão do modelo de árvore de decisão usando apenas os dados originais, que alcançou uma acurácia de quase 88%. Observe que 310 URLs fraudulentas foram classificadas incorretamente. À direita, após adicionar as features extraídas da URL, a acurácia do mesmo algoritmo subiu para mais de 93%, e o número de falsos negativos caiu para 232. Isso demonstra a importância da informação contida na própria estrutura da URL.

ML Clássico: Resultados dos Algoritmos

Table 1: Métricas para a classe "Fraudulenta".

Algoritmo de ML	Precisão (%)	Recall (%)	F1-score (%)
Regressão Logística	16	75	27
SVC (Sigmoid)	6	59	11
SVC (Poly)	55	93	69
SVC (RBF)	68	94	79
Árvore de Decisão	**89**	**91**	**90**

- Foram testados: Regressão Logística, Árvore de Decisão, Support Vector Machine (com diferentes kernels) e Redes Neurais.
- A **Árvore de Decisão** e o **SVM com kernel RBF** mostraram os melhores resultados em termos de Recall e F1-score, métricas cruciais para este problema.

ML Clássico: Resultados dos Algoritmos
Table 1: Métricas para a classe "Fraudulenta".
Algoritmo de ML | Precisão (%) | Recall (%) | F1-score (%)
Regressão Logística | 16 | 75 | 27
SVC (Sigmoid) | 6 | 59 | 11
SVC (Poly) | 55 | 93 | 69
SVC (RBF) | 68 | 94 | 79
Árvore de Decisão | 89 | 91 | 90
• Foram testados: Regressão Logística, Árvore de Decisão, Support Vector Machine (com diferentes kernels) e Redes Neurais.
• A Árvore de Decisão e o SVM com kernel RBF mostraram os melhores resultados em termos de Recall e F1-score, métricas cruciais para este problema.

Vários algoritmos clássicos foram testados. Esta tabela foca no mais importante: a capacidade de identificar corretamente as URLs fraudulentas. Em cibersegurança, o Recall é vital, pois queremos minimizar os falsos negativos – ou seja, URLs maliciosas que o sistema deixa passar. Vemos que a Regressão Logística e o SVM com kernel Sigmoid tiveram um desempenho ruim em precisão. Já o SVM com kernels Polinomial e RBF, e especialmente a Árvore de Decisão, alcançaram excelentes resultados, com F1-scores de 79% e 90%, respectivamente. O F1-score é uma boa medida geral porque equilibra precisão e recall.

ML Clássico: Redes Neurais e Modelos Ótimos

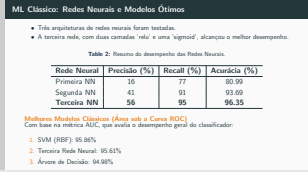
- Três arquiteturas de redes neurais foram testadas.
- A terceira rede, com duas camadas 'relu' e uma 'sigmoid', alcançou o melhor desempenho.

Table 2: Resumo do desempenho das Redes Neurais.

Rede Neural	Precisão (%)	Recall (%)	Acurácia (%)
Primeira NN	16	77	80.99
Segunda NN	41	91	93.69
Terceira NN	**56**	**95**	**96.35**

Melhores Modelos Clássicos (Área sob a Curva ROC)
Com base na métrica AUC, que avalia o desempenho geral do classificador:

1. SVM (RBF): 95.86%
2. Terceira Rede Neural: 95.61%
3. Árvore de Decisão: 94.98%



Os autores também exploraram redes neurais. A terceira configuração, que usava uma arquitetura um pouco mais complexa, superou as outras duas de forma expressiva, atingindo um Recall de 95% e uma acurácia de mais de 96%. Para fazer um resumo final dos modelos clássicos, a métrica AUC, ou Área sob a Curva ROC, é excelente. Ela nos dá um valor único que representa a capacidade do modelo de distinguir entre as classes. Com base nisso, o SVM com kernel RBF foi o campeão, seguido de perto pela terceira rede neural e pela árvore de decisão. Todos apresentaram um desempenho de altíssimo nível.

A Fronteira Quântica: Quantum Machine Learning (QML)

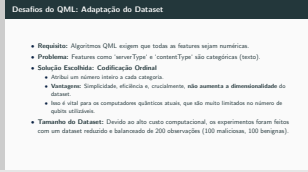
- O QML representa uma nova abordagem para resolver problemas de ML.
- Este trabalho foca na abordagem **CQ**: dados **C**lássicos processados por algoritmos quânticos em hardware **Q**uântico (simulado).
- **Modelo Utilizado: Variational Quantum Classifier (VQC)**
  - Uma variante quântica de uma rede neural.
  - Usa um circuito quântico parametrizado. Os "pesos" do modelo são os parâmetros do circuito, que são otimizados durante o treinamento.
  - Desenvolvido em Python com a framework **Qiskit** da IBM.

A Fronteira Quântica: Quantum Machine Learning (QML)
- O QML representa uma nova abordagem para resolver problemas de ML.
- Este trabalho foca na abordagem CQ: dados Clássicos processados por algoritmos quânticos em hardware Quântico (simulado).
- Modelo Utilizado: Variational Quantum Classifier (VQC)
  - Uma variante quântica de uma rede neural.
  - Usa um circuito quântico parametrizado. Os "pesos" do modelo são os parâmetros do circuito, que são otimizados durante o treinamento.
  - Desenvolvido em Python com a framework Qiskit da IBM.

Agora, entramos na parte mais inovadora do estudo: o Quantum Machine Learning. O QML busca utilizar os princípios da mecânica quântica, como superposição e emaranhamento, para realizar cálculos de aprendizado de máquina. A abordagem adotada aqui é a "CQ", onde dados clássicos do nosso problema são codificados em um estado quântico e processados por um algoritmo quântico, que neste caso foi simulado em um computador clássico. O modelo específico foi o Classificador Quântico Variacional, ou VQC. Pense nele como um análogo quântico de uma rede neural, onde em vez de ajustar pesos, otimizamos os parâmetros de um circuito quântico para classificar os dados.

Desafios do QML: Adaptação do Dataset

- **Requisito:** Algoritmos QML exigem que todas as features sejam numéricas.
- **Problema:** Features como 'serverType' e 'contentType' são categóricas (texto).
- **Solução Escolhida: Codificação Ordinal**
  - Atribui um número inteiro a cada categoria.
  - **Vantagens:** Simplicidade, eficiência e, crucialmente, **não aumenta a dimensionalidade** do dataset.
  - Isso é vital para os computadores quânticos atuais, que são muito limitados no número de qubits utilizáveis.
- **Tamanho do Dataset:** Devido ao alto custo computacional, os experimentos foram feitos com um dataset reduzido e balanceado de 200 observações (100 maliciosas, 100 benignas).



Aplicar QML não é trivial. O primeiro grande obstáculo é a preparação dos dados. Os algoritmos quânticos precisam de entradas puramente numéricas. Variáveis de texto, como o tipo de servidor, precisavam ser convertidas. Os autores optaram pela codificação ordinal, que é simples e, o mais importante, não cria novas colunas, mantendo a dimensionalidade baixa. Isso é uma consideração crítica, pois os processadores quânticos de hoje têm um número muito limitado de qubits. Outra limitação é o tempo de processamento. Treinar modelos quânticos é extremamente lento, então os autores tiveram que usar uma amostra bem pequena e balanceada do dataset para os experimentos quânticos.

QML: Configuração Experimental

Foram testadas diversas combinações de três componentes principais do VQC:

1. **Feature Map:** Algoritmo que codifica os dados clássicos em estados quânticos.
   - *Testados: ZZFeatureMap, ZFeatureMap, PauliFeatureMap.*
2. **Ansatz:** O circuito quântico parametrizado que "aprende".
   - *Testados: RealAmplitudes, EfficientSU2, ExcitationPreserving.*
3. **Otimizador:** Algoritmo clássico que ajusta os parâmetros do Ansatz.
   - *Testados: COBYLA, GradientDescent, SLSQP.*

QML: Configuração Experimental

Foram testadas diversas combinações de três componentes principais do VQC.

1. **Feature Map:** Algoritmo que codifica os dados clássicos em estados quânticos.
   - *Testados: ZZFeatureMap, ZFeatureMap, PauliFeatureMap.*
2. **Ansatz:** O circuito quântico parametrizado que "aprende".
   - *Testados: RealAmplitudes, EfficientSU2, ExcitationPreserving.*
3. **Otimizador:** Algoritmo clássico que ajusta os parâmetros do Ansatz.
   - *Testados: COBYLA, GradientDescent, SLSQP.*

A experimentação com QML envolveu testar sistematicamente diferentes combinações dos três blocos de construção de um VQC. O "Feature Map" é a maneira como os dados de entrada são "carregados" no circuito quântico. O "Ansatz" é o próprio circuito variacional, o coração do modelo de aprendizado. E o "Otimizador" é um algoritmo clássico que guia o processo de treinamento, ajustando os parâmetros do Ansatz para minimizar uma função de custo. O objetivo era encontrar a combinação que fornecesse a melhor precisão.

QML: Resultados e Destaques

A combinação de parâmetros com melhor desempenho se destacou:

- **Feature Map:** `ZFeatureMap`
- **Ansatz:** `RealAmplitudes`
- **Otimizador:** `SLSQP`

Resultados com a melhor combinação:

- Score no treino: 0.89
- Score no teste: **0.97**

Comparação com SVM Clássico (no mesmo dataset reduzido):

- Score no teste: **0.97**

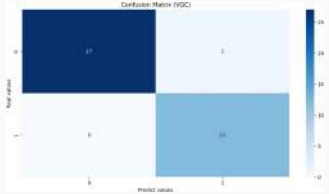
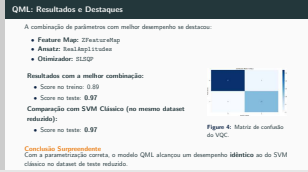


Figure 4: Matriz de confusão do VQC.

Conclusão Surpreendente

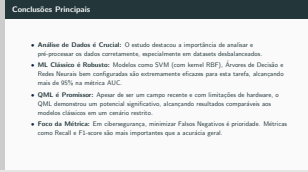
Com a parametrização correta, o modelo QML alcançou um desempenho **idêntico** ao do SVM clássico no dataset de teste reduzido.



E aqui estão os resultados da exploração quântica. A combinação do feature map ZFeatureMap, o ansatz RealAmplitudes e o otimizador SLSQP produziu os melhores resultados. Como podem ver, no conjunto de teste, este modelo VQC alcançou um score de 0.97. O mais impressionante é que, ao rodar o algoritmo SVM clássico no mesmo dataset reduzido, o resultado foi exatamente o mesmo: 0.97. A matriz de confusão para o VQC mostra que, dos 40 exemplos de teste, ele errou apenas 1. Isso é um resultado muito promissor, indicando que, mesmo em seu estágio inicial e com recursos limitados, o QML pode atingir um desempenho competitivo com os melhores métodos clássicos.

Conclusões Principais

- **Análise de Dados é Crucial:** O estudo destacou a importância de analisar e pré-processar os dados corretamente, especialmente em datasets desbalanceados.
- **ML Clássico é Robusto:** Modelos como SVM (com kernel RBF), Árvores de Decisão e Redes Neurais bem configuradas são extremamente eficazes para esta tarefa, alcançando mais de 95% na métrica AUC.
- **QML é Promissor:** Apesar de ser um campo recente e com limitações de hardware, o QML demonstrou um potencial significativo, alcançando resultados comparáveis aos modelos clássicos em um cenário restrito.
- **Foco da Métrica:** Em cibersegurança, minimizar Falsos Negativos é prioridade. Métricas como Recall e F1-score são mais importantes que a acurácia geral.

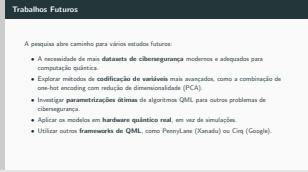


Para resumir as conclusões. Primeiro, a importância de uma análise de dados cuidadosa não pode ser subestimada. Segundo, os modelos de Machine Learning clássicos são muito poderosos e maduros, oferecendo soluções robustas e de alto desempenho para a detecção de URLs maliciosas. Terceiro, e talvez o mais excitante, é que o QML, mesmo sendo uma tecnologia incipiente, já se mostra muito promissor. Ele conseguiu igualar o desempenho de um modelo clássico forte, o que abre portas para pesquisas futuras à medida que o hardware quântico evolui. Por fim, o artigo reforça que, no domínio da cibersegurança, a escolha da métrica de avaliação é fundamental; é preferível ter alguns falsos positivos do que deixar uma ameaça real passar despercebida.

Trabalhos Futuros

A pesquisa abre caminho para vários estudos futuros:

- A necessidade de mais **datasets de cibersegurança** modernos e adequados para computação quântica.
- Explorar métodos de **codificação de variáveis** mais avançados, como a combinação de one-hot encoding com redução de dimensionalidade (PCA).
- Investigar **parametrizações ótimas** de algoritmos QML para outros problemas de cibersegurança.
- Aplicar os modelos em **hardware quântico real**, em vez de simulações.
- Utilizar outros **frameworks de QML**, como PennyLane (Xanadu) ou Cirq (Google).



Este trabalho não é o fim da linha, mas sim um ponto de partida. Os autores identificaram várias direções para pesquisas futuras. Há uma carência de datasets de cibersegurança de alta qualidade, que será crucial para treinar modelos mais avançados. Há espaço para explorar técnicas de codificação de dados mais sofisticadas. Conforme a tecnologia avança, será fundamental testar esses modelos em hardware quântico real e explorar outros frameworks de software que estão sendo desenvolvidos por empresas como Google e Xanadu. Este estudo, portanto, estabelece uma base sólida para a integração de algoritmos quânticos na detecção precoce de ações fraudulentas.

Obrigado!

Perguntas?

Obrigado!

Perguntas?

Gostaria de agradecer a todos pela atenção e abrir para perguntas. Também gostaria de repassar os agradecimentos dos autores originais às instituições que apoiaram esta importante pesquisa.