MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO

DANIEL CAPELLO CARVALHO

ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS NA IDENTIFICAÇÃO DE PADRÕES EM CRIPTOGRAMAS GERADOS POR SISTEMAS DE CRIPTOGRAFIA ASSIMÉTRICA

RIO DE JANEIRO
2021

DANIEL CAPELLO CARVALHO

ALGORITMOS DE APRENDIZADO DE MÁQUINA APLICADOS NA IDENTIFICAÇÃO DE PADRÕES EM CRIPTOGRAMAS GERADOS POR SISTEMAS DE CRIPTOGRAFIA ASSIMÉTRICA

Dissertação apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciência em Sistemas e Computação.

Orientador(es): José Antonio Moreira Xexéo, D.Sc.
Renato Hidaka Torres, D.Sc.

Rio de Janeiro

2021

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro – RJ CEP: 22290-270

Carvalho, Daniel Capello.
Algoritmos de aprendizado de máquina aplicados na identificação de padrões em criptogramas gerados por sistemas de criptografia assimétrica / Daniel Capello Carvalho. – Rio de Janeiro, 2021.
83 f.

Orientador(es): José Antonio Moreira Xexéo e Renato Hidaka Torres.

Dissertação (mestrado) – Instituto Militar de Engenharia, Sistemas e Computação, 2021.

1. Criptografia Assimétrica. 2. Aprendizado de Máquina. 3. Identificação de Padrões. 4. Ataque de Distinção. i. Xexéo, José Antonio Moreira (orient.) ii. Torres, Renato Hidaka (orient.) iii. Título

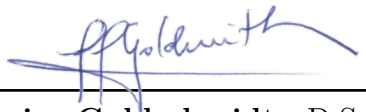
DANIEL CAPELLO CARVALHO

Algoritmos de aprendizado de máquina aplicados na identificação de padrões em criptogramas gerados por sistemas de criptografia assimétrica

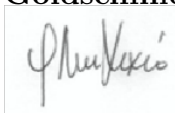
Dissertação apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciência em Sistemas e Computação.

Orientador(es): José Antonio Moreira Xexéo e Renato Hidaka Torres.

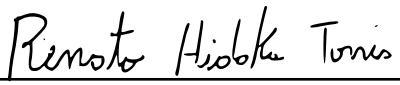
Aprovado em Rio de Janeiro, 26 de agosto de 2021, pela seguinte banca examinadora:



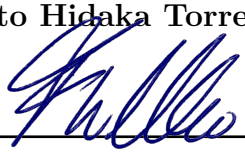
Prof. **Ronaldo Ribeiro Goldschmidt** - D.Sc. do IME - Presidente



Prof. **TC Rfm José Antônio Moreira Xexéo** - D.Sc. do IME



Prof. **Renato Hidaka Torres** - D.Sc. do UFPA



Prof. **Flavio Luis de Mello** - D.Sc. do UFRJ

Rio de Janeiro

2021

Este trabalho é dedicado à minha esposa Ingrid.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo dom da vida.

Agradeço a meus pais Ângela e Pedro, que dedicaram suas vidas aos filhos, oferecendo sempre o melhor que podiam.

Além de dedicar este trabalho, também agradeço à minha esposa Ingrid pelo companheirismo e apoio em todos os momentos difíceis que passei nessa jornada.

Agradeço aos professores doutores José Antônio Moreira Xexéo e Renato Hidaka Torres pela paciência que tiveram comigo e por terem me guiado pelos passos da pesquisa científica.

Agradeço aos colegas de curso Magno Alves Cavalcante e Felipe de Lima e Lima pelo companheirismo, apoio nas disciplinas e condução da minha pesquisa.

Por último, mas não menos importante, a todos os professores do Instituto Militar de Engenharia de quem eu tive o privilégio de ser aluno.

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke

RESUMO

Atualmente a comunicação segura na internet é baseada no protocolo *Transport Layer Security*. Neste protocolo, servidor e máquina cliente autenticam-se mutuamente e negociam o algoritmo de criptografia e as chaves criptográficas. Nesta etapa inicial, a criptografia assimétrica é central para as tarefas de autenticação e troca de chaves. *Ciphertext only attack* é um tipo de ataque utilizado na criptoanálise e é considerado bem-sucedido se qualquer informação do texto cifrado ou da chave puderem ser deduzidas. Uma das mais importantes tarefas na criptoanálise neste tipo de ataque é a identificação do algoritmo de criptografia utilizado. Pelo estudo dos autores anteriores, foram identificados poucos trabalhos que analisaram criptogramas oriundos de cifras assimétricas e nenhum deles avaliaram exclusivamente criptogramas assimétricos. Portanto, este trabalho objetivou avaliar a ocorrência de padrões em criptogramas gerados pelos algoritmos RSA, Elgamal e Curvas Elípticas utilizando uma abordagem supervisionada com os algoritmos de aprendizado *Support Vector Machine*, *Random Forest* e *K-Nearest Neighbor*. Como contribuições desta pesquisa, destacam-se a detecção de padrões em um estudo exclusivo de cifras assimétricas, em um contexto de múltiplas chaves, a análise de padrões em criptogramas ElGamal, ainda não avaliado por outros autores e a análise da influência do tamanho do criptograma nas medidas de desempenho dos classificadores. Os resultados alcançados mostraram a existência de padrões nos criptogramas e o melhor desempenho do classificador SVM. Foi possível identificar os criptogramas das Curvas Elípticas com acurácia de até 80% nos cenários comparativos com os demais criptogramas e não foi possível distinguir os criptogramas RSA e Elgamal quando são comparados um contra o outro. Também foi possível verificar o mínimo de 50 mil *bits* para uma acurácia maior que um acerto aleatório.

Palavras-chave: Criptografia Assimétrica. Aprendizado de Máquina. Identificação de Padrões. Ataque de Distinção.

ABSTRACT

Currently, secure internet communication is based on the Transport Layer Security protocol. In this protocol, server and client machine authenticate each other and negotiate the encryption algorithm and cryptographic keys. In this initial step, asymmetric encryption is central to the tasks of authentication and key exchange. Ciphertext only attack is a type of attack used in cryptanalysis and is considered successful if any ciphertext or key information can be deduced. One of the most important cryptanalysis tasks in this type of attack is the identification of the used encryption algorithm. From the study of previous authors, few studies were identified that analyzed cryptograms from asymmetric ciphers and none of them exclusively evaluated asymmetric cryptograms. Therefore, this work aimed to evaluate the occurrence of patterns in cryptograms generated by the RSA, Elgamal and Elliptic Curves algorithms using a supervised approach with the Support Vector Machine, Random Forest and K-Nearest Neighbor learning algorithms. As contributions of this research, we highlight the detection of patterns in an exclusive study of asymmetric ciphers, in a context of multiple keys, the analysis of patterns in ElGamal cryptograms, not yet evaluated by other authors, and the analysis of the influence of cryptogram size on the performance measures of the classifiers. The results achieved showed the existence of patterns in the cryptograms and the best performance of the SVM classifier. It was possible to identify the Elliptic Curves cryptograms with accuracy of up to 80% in the comparative scenarios with the other cryptograms and it was not possible to distinguish the RSA and Elgamal cryptograms when they are compared against each other. It was also possible to verify the minimum of 50 thousand bits for an accuracy greater than a random hit.

Keywords: Asymetric Cryptography. Machine Learning. Pattern Identification. Distinguishing Attack.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACC	Acurácia
ACM	*Association for Computing Machinery*
AES	*Advanced Encryption Standard*
CE	Curvas Elípticas
DES	*Data Encryption Standard*
ECDLP	*Elliptic Curve Discrete Logarithm Problem*
ELG	Elgamal
FN	Falso Negativo
FP	Falso Positivo
F1	Média harmônica entre Precisão e *Recall*
GCHQ	*UK Government Communication Headquarter*
IACR	*International Association for Cryptologic Research*
IDEA	*International Data Encryption Algorithm*
IEEE	*Institute of Electrical and Electronics Engineers*
IETF	*Internet Engineering Task Force*
KNN	*K-Nearest Neighbor*
NSA	*National Security Agency*
PKCS	*Public Key Cryptography Standards*
PV	Positivo Verdadeiro
NV	Negativo Verdadeiro
RSA	Rivest Shamir Adleman (Sobrenomes dos criadores do algoritmo)
RF	*Random Forest*
SVM	*Support Vector Machine*
TLS	*Transport Layer Security*

LISTA DE SÍMBOLOS

C	Constante utilizada no classificador SVM
w	Vetor que determina a fronteira de decisão
α	Significância da inferência do experimento
β	Complementar da potência do experimento
δ	Erro experimental
γ	Parâmetro utilizado na função Kernel *Radial Basis Function*
$\phi(n)$	Função Totiente de Euler do parâmetro n
σ	Desvio padrão
Zα	Variável normal padronizada para α
Zβ	Variável normal padronizada para β

SUMÁRIO

1 INTRODUÇÃO

A confidencialidade como um princípio ético está associada à conduta nas relações humanas em diversas profissões, como por exemplo, medicina, direito, psicologia e jornalismo. No contexto da ciência da informação, é a propriedade pela qual o significado de uma determinada informação não estará disponível a indivíduos sem autorização para acessá-lo (4). Para garantir essa propriedade, diversos algoritmos e protocolos foram desenvolvidos. Atualmente, esses algoritmos de criptografia são ferramentas centrais para a confidencialidade de comunicações governamentais, militares, empresarias e pessoais.

Segundo Schneier(5), se a segurança de um algoritmo criptográfico é baseada no segredo de como ele funciona, então estas cifras enfrentam os seguintes problemas: 1) toda vez que um membro do grupo que se comunica pela cifra sair ou se alguém acidentalmente revelar seu funcionamento, será necessário modificá-lo; 2) a verificação de qualidade só será possível com as pessoas deste mesmo grupo; 3) cada grupo de usuários deverá desenvolver seu próprio algoritmo. Os algoritmos modernos de criptografia resolveram esse cenário problemático por meio do uso de chaves. Segundo o princípio de Kerckhoff(6), a fonte do segredo de um sistema criptográfico deverá residir em sua chave.

Uma chave criptográfica é um parâmetro utilizado em conjunto com um algoritmo criptográfico que determina seu funcionamento de tal forma que uma entidade com conhecimento da chave pode reproduzir, reverter ou verificar a operação, enquanto uma entidade sem conhecimento da chave não consegue realizá-la, em um tempo computacionalmente aceitável (7). Desta forma, cada membro de um grupo poderia compartilhar a mesma chave para trocar mensagens sigilosas entre si, e um segundo grupo poderia ter uma chave diferente, porém ambos os grupos poderiam compartilhar o mesmo algoritmo.

Se por um lado, tornar o algoritmo criptográfico público e manter em segredo apenas a chave resolve os problemas citados acima, ele possibilita aos criptoanalistas um estudo mais profundo de possíveis vulnerabilidades. Estas vulnerabilidades residem tanto nos modelos teóricos utilizados pela criptografia, quanto nas deficiências em suas implementações. Até o aspecto humano de uso destes sistemas são explorados a fim de se obter informações sobre a mensagem ou chave.

Ciphertext only attack, ou ataque de textos cifrados, é um tipo de ataque utilizado na criptoanálise e é considerado bem-sucedido se qualquer informação do texto cifrado ou da chave puderem ser deduzidas. O processo de seleção de um novo padrão de algoritmo criptográfico geralmente leva muitos anos e inclui testes exaustivos de grandes quantidades de texto cifrado para qualquer abordagem estatística não detectar informação relevante, além de ruído aleatório (8). Dileep e Sekhar(9) destacam que as duas mais importantes

tarefas na criptoanálise em que se possui apenas textos cifrados são: 1) A identificação do algoritmo de criptografia e 2) a identificação da chave utilizada.

A identificação do algoritmo de criptografia, denominado ataque de distinção por Knudsen e Meier(10), foi estudado por diversos autores desde 2001, como será apresentado no Capítulo 2 e Anexo A. De acordo com o trabalho de revisão bibliográfica realizado nesta pesquisa, mais de 80% dos trabalhos identificados, avaliaram os algoritmos criptográficos simétricos. Poucos estudos incluíram em sua análise algoritmos assimétricos. Diante disso, esta pesquisa localiza-se no conxtexto de ataque de distinção em cifras assimétricas.

1.1 Motivação

Atualmente a comunicação segura na internet é baseada no protocolo *Transport Layer Security*. Na primeira etapa deste protocolo, chamada de *TLS Handshake*, servidor e máquina cliente autenticam-se mutuamente e negociam o algoritmo de criptografia e as chaves criptográficas antes que a segunda parte do protocolo transmita ou receba seu primeiro *byte* de dados da mensagem (11). Nesta etapa inicial, a criptografia assimétrica é central para as tarefas de autenticação e troca de chaves. Nas etapas posteriores, a mensagem que se quer proteger é encriptada por algoritmos simétricos, utilizando a chave simétrica trocada na primeira parte. Este modelo atual é chamado de híbrido, uma vez que se utiliza algoritmos assimétricos e simétricos.

Dada a centralidade da criptografia assimétrica, a pouca produção científica relacionada a ataques de distinção para esse conjunto de cifras, como será identificada no capítulo 2, é uma lacuna importante a ser tratada e que motiva este trabalho. Mesmo que a quantidade de *bits* encriptados por algoritmos assimétricos seja pequena (troca de chaves), comparada ao tamanho da mensagem no protocolo TLS, em um ataque de texto cifrado, saber qual a cifra assimétrica utilizada já é um passo importante para a criptoanálise. Este cenário também leva a uma indagação do menor volume de dados necessários para uma possível identificação da cifra criptográfica, que será abordada como resultado complementar desta pesquisa.

A classificação de criptogramas pelos algoritmos que o geraram é um exemplo de tarefa que está na interseção de duas grandes áreas da ciência da computação: Criptologia e Inteligência Artificial. Avanços nessa fronteira de conhecimento possibilitam externalidades positivas para diversas áreas destas disciplinas, tais como automação em defesa cibernética e desenvolvimento de novos modelos de algoritmos de criptografia. No Capítulo 2, identifica-se uma oportunidade de estudo para investigar padrões em criptogramas de cifras assimétricas por meio dos algoritmos de aprendizado *K-Nearest Neighbor*, *Random Forest* e *Support Vector Machines*, utilizados por diversos autores até então para tratar apenas das cifras simétricas.

É importante mencionar também o histórico de trabalhos gerados pelo grupo de segurança da informação do Instituto Militar de Engenharia (GSI/IME) sobre esta temática. Desde 2006, diversos trabalhos foram desenvolvidos gerando uma massa crítica sobre ataques de distinção. Contribuir para este legado e o aprimoramento desta especialidade dentro do GSI/IME é um fator motivacional para este autor.

1.2 Caracterização do problema

A tarefa de identificação do algoritmo criptográfico significa buscar soluções para o seguinte problema:

Dado um criptograma $c_i = f(text_i, alg_j, key_k)$, resultado da encriptação de um texto em claro ($text_i$) por um algoritmo criptográfico (alg_j), utilizando uma chave (key_k), é possível descobrir alg_j através da detecção de alguma informação contida em c_i?

Os trabalhos anteriores estudaram este problema sob diversas abordagens, tais como recuperação de informações, utilização de técnicas de *data mining* e sob a ótica do reconhecimento de padrões (*pattern recognition*), utilizando classificadores não supervisionados e supervisionados. Esta pesquisa utilizou a abordagem de reconhecimento de padrões com classificadores supervisionados. O reconhecimento de padrões é formalmente descrito desta forma: dada uma função desconhecida $g : X \rightarrow Y$, que mapeia elementos $x \in X$ em rótulos $y \in Y$. Presumindo-se que o conjunto de dados $D = (x_1, y_1), \ldots, (x_n, y_n)$, que são instâncias representativas deste mapeamento, produzem uma função $h : X \rightarrow Y$ que se aproxima o máximo possível da função original g, esta "aproximação" é definida como a minimização do valor especificado por alguma função de perda/custo que atribui um valor à classificação errada em rótulo do contra-domínio Y. Esta abordagem ao problema também será adotada por este trabalho.

A Figura 1 procura descrever visualmente o contexto do problema. Dado um conjunto de 30.279 textos, cada um é criptografado pelos algoritmos RSA, ElGamal e Curvas Elípticas, utilizando diferentes chaves k. Dado que esses criptogramas compõem uma base e supondo que há uma função hipotética $g : C \rightarrow A$ que mapeia o domínio dos criptogramas $c_i \in C$ no contra-domínio formado pelos tipos de algoritmos $alg_j \in A$. O objetivo é extrair informações e características de c_i que possam construir uma função h, de maneira supervisionada, através de algoritmos de aprendizado de máquina, que seja a mais próxima possível de g (ou seja, que minimize os erros de classificação dos criptogramas), para classificar um novo criptograma.

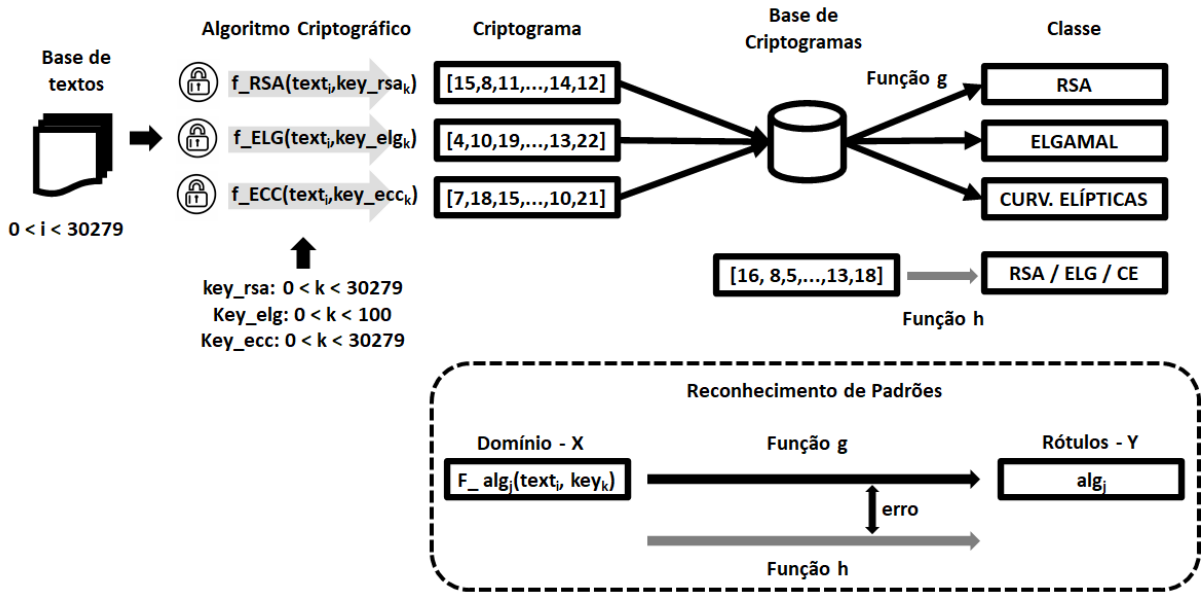


Figura 1 – Ilustração esquemática do problema de pesquisa.

1.3 Hipótese

Hipótese: Existem padrões nos criptogramas gerados pelo algoritmo ElGamal que podem ser identificados pelos algoritmos de aprendizado de máquina.

Conforme estudos anteriores, apresentados no capítulo 2, foram identificados padrões nos criptogramas RSA e Curvas elípticas. Portanto, esta hipótese extrapola a identificação de padrões para o ElGamal. A hipótese não será válida se os algoritmos de aprendizado de máquina não conseguirem distinguir os criptogramas de acordo com o seu algoritmo criptográfico. Isto significa uma acurácia igual a de um acerto aleatório.

1.4 Objetivo Geral

O objetivo desta pesquisa foi avaliar a ocorrência de padrões em criptogramas gerados pelos algoritmos assimétricos RSA, ElGamal e Curvas Elípticas de forma que seja possível classificar um novo criptograma utilizando os algoritmos de aprendizado SVM, *Random Forest* e *K-Nearest Neighbor*.

1.5 Resultados Complementares

Como resultado complementar estudado nesta pesquisa, será investigado o comportamento do desempenho dos classificadores à medida em que a quantidade de *bits* é alterada, objetivando identificar a quantidade mínima de bits necessários para uma classificação pelos algoritmos de aprendizado com acurácia maior que um acerto aleatório.

1.6 Método de pesquisa

Segundo Wazlawick(12), o método de pesquisa consiste na sequência de passos necessários para demonstrar que o objetivo foi atingido. O método utilizado nesta pesquisa compreende as seguintes etapas:

1. **Escolha do Tema e problema geral de pesquisa:** Conforme descrito na Motivação 1.1 deste trabalho, a identificação do algoritmo criptográfico a partir dos criptogramas é um tema já pesquisado no IME. Este tema foi adotado com o objetivo de expandir os resultados acumulados pelos pesquisadores anteriores;

2. **Estudo dos trabalhos anteriores e identificação da abordagem utilizada:** Os autores identificados nesse estudo utilizaram algoritmos de aprendizado de máquinas e técnicas de agrupamento, aliadas com técnicas de recuperação de informações, para a solução do problema de identificação do algoritmo criptográfico. Nesta etapa do método, são estudadas as formas de abordagem ao problema e as limitações de cada estudo.

3. **Identificação da lacuna de pesquisa:** Como mostrado no Capítulo 2, diversos autores estudaram o problemas. Em sua maioria, os estudos identificados avaliaram cifras provenientes de algoritmos criptográficos simétricos. Apenas os trabalhos de Souza(13), Oliveira, Souza e Xexeo(14), Torres et al.(15), Manjula e Anitha(16), Mello e Xexeo(17), Barbosa, Vidal e Mello(18), Barbosa et al.(19) e Mello e Xexeo(20) abordaram também os algoritmos assimétricos em seus experimentos. Barbosa(21) destaca o problema de fatoração de grandes números (RSA), logaritmo discreto em corpos finitos (ElGamal) e logaritmo discreto em curvas elípticas (ECDLP) como os principais problemas nos quais os sistemas de criptografia assimétrica se baseiam atualmente. Portanto, ainda há uma lacuna de investigação em outros algoritmos assimétricos além do RSA e Curvas Elípticas (16) e o uso de algoritmos de aprendizado ainda não aplicados ao contexto de cifras assimétricas (ver Tabela 3). Tal lacuna de estudo é corroborada também por (22), que considera a avaliação de padrões em criptogramas gerados por algoritmos assimétricos como sugestão de trabalho futuro.

4. **Delimitação do problema e determinação da hipótese e objetivo**: Uma vez identificada a lacuna de pesquisa, foram delimitados o problema a ser resolvido, definidos a hipótese a ser testada e o objetivo de pesquisa.

5. **Planejamento e implementação dos experimentos:** Primeiramente, uma base de textos em inglês foi criptografada, com diversas chaves diferentes, nos algoritmos RSA, ElGamal e Curvas Elípticas. Assim como o processamento de linguagem natural, onde textos são representados por vetores que contam a quantidade das

palavras presentes, os criptogramas são vetorizados da mesma forma, onde cada palavra é uma sequência de 8 bits. Os vetores são classificados de acordo com o algoritmo criptográfico e compõem a base de entrada dos algoritmos de aprendizado, sendo repartidos em conjuntos de treinamento, testes e validação. Caso os algoritmos consigam encontrar fronteiras de decisão que possibilitem distinguir os criptogramas, com uma acurácia maior que um acerto aleatório, tem-se a confirmação da hipótese desta pesquisa.

6. **Análise dos resultados e comparação com os demais autores:** Nesta etapa, os resultados obtidos são analisados e comparados com os autores identificados. Busca-se compreender as causas que levaram à obtenção dos resultados e suas diferenças para os demais trabalhos.

1.7 Organização da Dissertação

Como descrito na seção anterior, cada etapa do método de pesquisa está organizada em um capítulo. No Capítulo 2, complementado pelo Anexo A, é apresentada a revisão bibliográfica realizada nesta pesquisa, identificando como os diversos autores abordaram o tema de identificação de padrões em criptogramas, bem como a identificação da lacuna de pesquisa apresentada no objetivo e hipóteses deste capítulo.

No Capítulo 3, são abordadas as definições e os conceitos fundamentais dos algoritmos criptográficos e de aprendizado de máquina utilizados nesta pesquisa. Adicionalmente, as formas de avaliação e mensuração dos resultados são apresentadas e definidas.

No Capítulo 4, é descrito o detalhamento do método de pesquisa e sua implementação, bem como os resultados dos experimentos executados.

Por fim, no Capítulo 5, os resultados serão consolidados e são feitas comparações com os trabalhos anteriores identificados. Conclusões acerca das hipóteses e objetivos são discutidas, bem como caminhos para trabalhos futuros são comentados.

2 TRABALHOS RELACIONADOS

Este capítulo descreve a investigação dos trabalhos anteriores relacionados ao tema central desta pesquisa. O objetivo é analisar como os demais pesquisadores abordaram o problema de identificação de padrões em cifras assimétricas, o uso dos modelos de aprendizado de máquina nesse contexto e suas capacidades de predição e generalização de resultados. Pretende-se identificar também lacunas de uso dos algoritmos de aprendizado para justificar a abordagem deste trabalho.

2.1 Questões de pesquisa

Para a execução desta pesquisa, algumas perguntas foram definidas para orientar a seleção e o entendimento dos estudos:

Item	Critério de análise
A1	Quais foram as formas de abordar o problema central dessa pesquisa?
A2	Quais os algoritmos de aprendizado mais adotados?
A3	Quais os parâmetros de configuração dos algoritmos de aprendizado?
A4	Quais os tipos de dados utilizados pelos pesquisadores?
A5	Quais as bases de dados que estão sendo utilizadas nesses estudos?
A6	Como foram os experimentos conduzidos pelos autores? Utilizam uma abordagem similar a outros autores, ou criaram abordagens exclusivas?
A7	Quais as técnicas de pré-processamentos utilizadas nos dados?
A8	Quais as métricas adotadas para avaliar os modelos desenvolvidos?
A9	Quais foram os resultados alcançados em relação a cifras assimétricas?
A10	Quais as dificuldades encontradas?

Tabela 1 – Critérios para análise dos estudos

2.2 Seleção de Fontes

Os veículos de publicações acadêmicas que possuem pesquisas relacionadas ao tema de criptografia e que foram o ponto de partida da pesquisa encontram-se listados abaixo:

- *Institute of Electrical and Electronic Engineers* - IEEE
- *Association for Computing Machinery* - ACM
- *The International Association for Cryptologic Research* - IACR

- *Brazilian Journal of Information Security and Cryptography* - ENIGMA
- *Cryptologia Journal*
- *Journal of Discrete Mathematical Sciences and Cryptography*

2.3 Método de identificação dos trabalhos de pesquisa

O ponto de partida para a pesquisa dos trabalhos anteriores foi o estudo das pesquisas de mestrado conduzidas no IME. A partir daí, foram definidas palavras-chave que serviram como parâmetros para as buscas dos trabalhos identificados neste estudo: "*Pattern reconginition*", "*cipher classification*", "*Cipher identification*", "*Classification/Identification of encription algorithms*", "*Machine Learning approach*", "*Distinguishing Attack*". Os termos foram pesquisados, tanto na língua inglesa quanto na língua portuguesa.

Ao analisar os trabalhos encontrados nas bases listadas acima, suas referências bibliográficas também foram analisadas, de forma a completar o mapeamento dos estudos que trataram do tema.

Os trabalhos citados nesta pesquisa foram selecionados a partir deste método de busca e utilizando os filtros de:

- Presença dos termos de busca no título ou resumo;
- Avaliação da aderência ao objetivo de pesquisa, através da leitura do resumo;
- Citação de outros trabalhos.

2.4 Critérios de qualidade

Adicionalmente às questões descritas acima que orientam a análise dos trabalhos, os critérios a seguir serão utilizados para guiar esta pesquisa em relação à qualidade dos trabalhos analisados.

Item	Critério de qualidade
Q1	Os objetivos/questões de estudo estão definidos claramente?
Q2	O trabalho é adequadamente referenciado (apresenta trabalhos relacionados e baseia-se em modelos e teorias da literatura)?
Q3	Os objetivos ou questões do estudo são alcançados?
Q4	Existe um método ou um conjunto de métodos descrito para a realização do estudo?
Q5	O estudo descreve de forma clara e não ambígua os resultados?

Tabela 2 – Critérios para análise da qualidade dos estudos

2.5 Panorama dos estudos encontrados e identificação da lacuna de escopo de pesquisa

As pesquisas acerca do problema de identificação do algoritmo criptográfico através dos criptogramas iniciaram em 2001. A Figura 2 mostra que o assunto foi mais estudado apenas na década de 2010, o que é possível verificar pela maior quantidade de marcações a partir desta data. Por exemplo, em 2001, o trabalho de Maheshwari(23) estudou a identificação da cifra nos algoritmos DES e IDEA. Cada símbolo identificado na figura representa um ou mais trabalhos que trataram do algoritmo criptográfico, de acordo com a legenda abaixo da figura. A primeira linha representa a quantidade de trabalhos por ano, enquanto a última coluna representa a quantidade de trabalhos por algoritmos criptográficos.

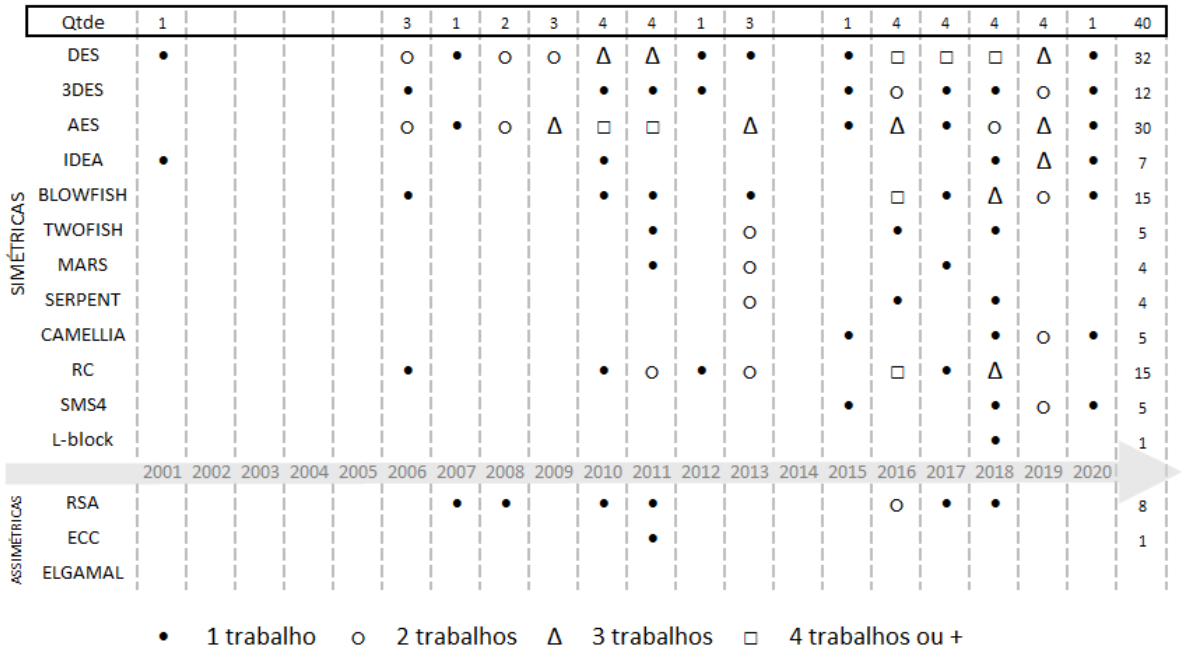


Figura 2 – Linha do Tempo de Algoritmos Criptográficos

Os trabalhos que estudaram os algoritmos assimétricos, que representam apenas 18% do total de trabalhos, têm inicio em 2007 (13). A Tabela 3 a seguir identifica os autores que trabalharam também com os algoritmos assimétricos e quais os algoritmos de aprendizado de máquina utilizados. Visualização semelhante também foi desenvolvida para os algoritmos simétricos e está apresentada no anexo A: tabelas 58, 59, 60 e 61.

Criptografia Assimétrica			
Algoritmo	RSA	ELGAMAL	CURVAS ELIPTICAS
Redes Neurais	(13) (14) (15) (18) (19) (20) (17)		
SVM	X	X	X
Árvore de Decisão	(16) (17) (20) (18) (19)		(16)
Random Forest	X	X	X
Naive Bayes	(18) (19) (20)		
Algoritmos Genéticos	(15)		
Grafos	(15)		
KNN	X	X	X

Tabela 3 – Classificadores x Algoritmos Criptográficos Assimétricos

Os trabalhos de Souza(13), Oliveira, Souza e Xexeo(14), Torres et al.(15), Manjula e Anitha(16), Mello e Xexeo(17), Barbosa, Vidal e Mello(18), Barbosa et al.(19) e Mello e Xexeo(20) estudaram ataques de distinção em cifras simétricas e assimétricas avaliando prioritariamente o RSA (exceto por (16) que abordou também as Curvas Elípticas). Ao comparar a Tabela 3 com as tabelas do anexo A, observa-se que nem todos os algoritmos de aprendizado foram utilizados nos estudos envolvendo cifras assimétricas e que não foi realizado nenhum estudo envolvendo apenas cifras assimétricas. Desta forma, as lacunas ainda não exploradas pelos pesquisadores, identificadas por este autor, são descritas nos tópicos abaixo:

- Inexistência de estudos que comparem exclusivamente cifras assimétricas;
- Carência de estudos que utilizem os algoritmos SVM, *Random Forest* e KNN na identificação de padrões em cifras assimétricas (assinalada com "X" na tabela 3);
- Inexistência de estudos que avaliem criptogramas gerados pelo algoritmo criptográfico ElGamal, que segundo Schneier(5) é um dos três algoritmos assimétricos eficientes tanto para a criptografia, quanto para o processo de assinatura digital, ao lado do RSA e Rabin.

O objetivo desta pesquisa é atender a estas lacunas identificadas.

2.6 Análises dos trabalhos

Ao analisar os trabalhos que trataram das cifras assimétricas pela ordem cronológica, a dissertação de mestrado de Souza(13) expande os estudos iniciados por Carvalho(24), incluindo o algoritmo RSA em sua gama de cifras. Ele também aborda o problema tratando os criptogramas como textos e utilizando técnicas de recuperação de informações combinadas com algoritmos de agrupamento hierárquico e redes neurais (mapa de Kohonen), seguindo uma abordagem não supervisionada.

Na etapa de pré-processamento, os criptogramas são representados por vetores que exprimem a frequência de blocos de bits, em uma aproximação da técnica de *bag-of-words*. Em seguida, foram calculadas a similaridade entre eles utilizando diversos índices. Vetores com similaridades próximas são agrupados, por meio da técnica hierárquica aglomerativa. Para avaliar o desempenho no agrupamento dos criptogramas, utilizou-se as medidas de precisão e *recall*. Foram executados 7 diferentes experimentos. Em um dos experimentos realizados, procura-se identificar os criptogramas de acordo com seus algoritmos (problema central desta pesquisa). Devido ao experimento comparar algoritmos de classes diferentes (simétrica e assimétrica), portanto, com chaves diferentes, seus resultados indicaram que, com a técnica de agrupamento hierárquico, o algoritmo criptográfico não gera características de associação nos criptogramas mais fortes do que as geradas por uma chave em particular, a ponto de um grupo ser formado apenas por criptogramas cifrados com um mesmo algoritmo criptográfico. O resultado quantitativo revelou uma precisão de 100%, porém com um *Recall* máximo de 2%.

Posteriormente foram publicados os experimentos envolvendo os agrupamentos pelas chaves em 2008 (14). Dentre as medidas de similaridades estudadas em (13), foi selecionada a medida do cosseno como a de melhor resultado para este estudo. Os mesmos tipos de dados e método foram adotados. Para o algoritmo RSA, a precisão máxima foi alcançada, porém com *Recall* de apenas 6% para textos com chaves de 512 bits e 7% para chaves com 1024 bits. Apesar de não ser um trabalho com a intenção de identificar o algoritmo criptográfico, como foram agrupados pelas chaves, e estas são características de cada algoritmo (DES, AES e RSA), pode-se indiretamente considerar também uma contribuição para esta pesquisa.

O próximo trabalho (15) congrega resultados dos trabalhos de mestrado (13), (8) e (25), ilustrando as diversas técnicas até então desenvolvidas para a separação dos criptogramas e repetindo os resultados alcançados no RSA apresentados em (14). Em Torres(8), além das contribuições para melhoria do algoritmo simétrico AES, é investigada a técnica de grafos para melhor visualização dos grupos de separação de criptogramas.

Em Manjula e Anitha(16), os autores utilizaram o algoritmo de aprendizado Árvore de Decisão (C4.5) para a identificação dos algoritmos criptográficos, em uma abordagem

de categorização de textos. Diferentemente da representação dos vetores pela frequência de blocos de bits, os autores utilizaram medidas de entropia dos diferentes símbolos dos criptogramas e outras medidas a saber: 1) Entropia máxima de todos os caracteres; 2) Entropia máxima das letras maiúsculas; 3) Entropia máxima das letras minúsculas; 4) Entropia máxima dos símbolos; 5) Entropia máxima dos números; 6) Entropia máxima de trigramas mais recorrentes; 7) Entropia máxima de quadrigramas mais recorrentes; 8) Entropia de todos os caracteres; 9) Coeficiente de correlação das letras maiúsculas; 10) Tamanho do criptograma.

Foi utilizado o método chi-quadrado (χ^2) para selecionar as 8 características de maior significância. Esta forma de pré-processamento difere das formas encontradas nos demais autores, configurando uma abordagem exclusiva. Até então, os criptogramas eram transformados em vetores que representam os histogramas de blocos de bits e eram calculadas as similaridades entre os documentos. Estas similaridades são utilizadas como entrada nos algortmos e técnicas citados pelos demais trabalhos até então.

Não foi destacada a fonte dos textos que foram encriptados, portanto não há uma forma de replicar os resultados alcançados. Como resultado para os criptogramas assimétricos RSA e Curvas Elípticas, foi possível identificar 73,2% das amostras RSA e 74,3% das Curvas Elípticas.

Os trabalhos (18) e (19) foram trabalhos publicados, em anos diferentes, pelos mesmos autores. A principal característica de ambas as pesquisas é a variedade de algoritmos de aprendizado de máquina estudados e os tipos diferentes de dados em claro. Em Barbosa, Vidal e Mello(18), foram utilizados textos, enquanto em Barbosa et al.(19), foram utilizados áudios e vídeos. Em ambos os trabalhos, os resultados são semelhantes: o melhor resultado alcançado pelos algoritmos utilizados na classificação de criptogramas RSA foi do algoritmo Bayesiano *Complement Naive Bayes*, onde alcançou 89,36% de acurácia, considerando uma representação dos criptogramas com blocos de 16 bits. Isso demonstra que, independentemente do tipo de arquivo inicial, os padrões que podem diferenciar criptogramas são oriundos dos algoritmos criptográficos.

Os estudos Mello e Xexeo(17) e Mello e Xexeo(20), também seguindo os resultados dos trabalhos apresentados acima, extrapolaram os cálculos para blocos maiores, utilizando mais recursos computacionais. O algoritmo *Complement Naive Bayes* ainda mostrou melhores resultados, alcançando acurácia próxima a 100% para blocos maiores de 30 bits. Ressalta-se que para esses estudos, é utilizada 1 chave de 1024 *bits* diferente para cada texto.

Ao consolidar os trabalhos analisados até o momento, sob a ótica das perguntas iniciais deste capítulo, as seguintes conclusões podem ser estabelecidas:

1. (A1) Abordagem ao problema: Para identificar padrões e consequentemente predizer

o algoritmo criptográfico gerador de um determinado criptograma, alguns autores utilizaram algoritmos não supervisionados para agrupamento dos criptogramas e outros utilizaram uma abordagem supervisionada, por meio do treinamento de algoritmos de aprendizado de máquinas. Pelos resultados apresentados, a abordagem supervisionada conseguiu atingir resultados melhores, principalmente de *Recall*, e esta será adotada para a condução desta pesquisa;

2. (A2) Algoritmos de aprendizado mais adotados: Ao analisar as tabelas mencionadas neste capítulo e nos anexos, os cinco algoritmos de aprendizado de máquina mais utilizados foram as Redes Neurais, *Support Vector Machines* - (SVM); Árvores de Decisão, Random Forest e Naive Bayes. Pelo levantamento de trabalhos identificados na tabela 3, os algoritmos SVM e Random Forest não foram utilizados para tratar de cifras assimétricas, apesar de serem bastante populares para o estudo das cifras simétricas (ver Anexo A). Por isso, estes algoritmos serão utilizados nos experimentos, conjuntamente com o KNN;

3. (A3) Parâmetros de Configuração dos algoritmos: Nenhum dos trabalhos que trataram das cifras assimétricas de forma supervisionada discutiram os parâmetros utilizados nos algoritmos de aprendizado. Não foi identificado também estudo de otimização destes parâmetros. Esta oportunidade de estudo será indicada na seção de trabalhos futuros;

4. (A4) Tipos de dados: Predominantemente, os textos são os tipos de dados mais utilizados,no entanto foram identificados trabalhos que abordaram também outros tipos de mídias (áudios e imagens). Serão utilizados textos como fontes de dados para esta pesquisa, de forma que seja possível a comparação com a maior parte dos trabalhos aqui identificados;

5. (A5) Base de dados: Não há uma base de criptogramas de referência para uniformização e possível comparação entre os trabalhos. Nem todos os trabalhos mencionaram qual base de dados foi utilizada. As bases de dados mencionadas foram: para textos em claro, a Bíblia e Reuters21578; Caltech-256 e CIFAR10 para imagens. Ainda foram estudados arquivos em áudio (19). Desta constatação, é sugerido na seção de trabalhos futuros um estudo específico para construção de uma base única para desenvolvimento de análises e estudos acerca do tema. Com relação ao tamanho da base de dados utilizada pelos autores, há também diferentes cenários: Oliveira, Souza e Xexeo(14) trabalharam com 30 textos de 10KB cada, Manjula e Anitha(16) utilizaram 2000 textos de diferentes tamanhos, Barbosa, Vidal e Mello(18) analisaram 200 textos com pelo menos 6.000 caracteres cada e Mello e Xexeo(17) utilizaram 600 amostras com pelo menos 140.000 caracteres. Para a procura de padrões em criptogramas, é necessária uma base com textos grandes, para que possíveis padrões

possam emergir. Desta forma, adotadou-se a premissa de utilizar uma grande base de textos com diferentes tamanhos, assim como (16). Mais informações sobre a base de dados utilizada neste trabalho estão detalhadas no capítulo 4;

6. (A6) Método dos experimentos: Não foram encontrados métodos muito diferentes dentre os autores que utilizaram a mesma abordagem (não-supervisionadas ou supervisionadas). Um aspecto importante na forma de condução do experimento é a quantidade de chaves utilizadas nos estudos. Nos trabalhos (13) e (14), foram utilizadas 50 chaves distintas para o algoritmo RSA. Cada chave foi utilizada para encriptação de todos os textos. Já nos trabalhos (17) e (20), foram utilizadas chaves distintas para cada texto em claro. Os demais trabalhos identificados não mencionam explicitamente como as chaves assimétricas foram utilizadas. Concordando com o argumento de possível introdução de viés (17), ao se utilizar um número pequeno de chaves, serão utilizadas chaves diferentes para cada texto. Em termos da separação dos dados entre treinamento e teste, todos os trabalhos que avaliaram cifras assimétricas, e mencionaram seus percentuais, utilizaram uma divisão 66% e 34% (18), (17), (19) e (20) ou 70% e 30% (16). Desta forma, também será adotada a divisão 66% e 34%, para possibilitar posterior comparação.

7. (A7) Nos trabalhos que utilizaram a abordagem não supervisionada, o pré-processamento padrão foi o cálculo das similaridades entre os criptogramas. No cenário de treinamento e teste dos algoritmos de aprendizado, a geração de vetores que mapeiam os blocos de bits do criptograma (*bag-of-words*) foi o método mais utilizado. A exceção encontrada para as cifras assimétricas encontra-se no trabalho de Manjula e Anitha(16), que utilizou diversas medidas de entropia em vez da frequência dos blocos de *bits*, conforme descrito anteriormente. Para melhor comparação, o pré-processamento utilizado neste estudo seguirá com a maioria dos autores, utilizando a frequência de *bits* encontrados;

8. (A8) Medida de Desempenho: Basicamente, alguns trabalhos apresentam como medidas de desempenho a combinação Precisão e *Recall* - derivando também o F1*score* enquanto outros trabalhos utilizam a Acurácia. A acurácia é uma medida geral de desempenho, porém não possibilita um entendimento sobre a taxa de acerto do classificador em uma determinada classe. Este detalhamento é alcançado pela utilização das medidas de precisão, *recall* e *F1-Score*. Por isso, as medidas de desempenho supracitadas serão utilizadas para descrever os resultados alcançados pelos algoritmos de aprendizado nos experimentos desta pesquisa;

9. (A9) Resultados alcançados: Como citado acima, para os métodos de agrupamento (não-supervisionados), alcançou-se a precisão ser de 100%, porém com um *Recall* baixo. Nos métodos supervisionados, os resultados de acurácia foram superiores

a 70%. Ressalta-se que todos os estudo trabalharam com algoritmos simétricos e assimétricos. A presença de criptogramas gerados pelos 2 tipos de algoritmos poderia reforçar a diferença de ambos, podendo aumentar a taxa de acerto. Por isso, esta pesquisa investigará apenas cifras assimétricas;

10. (A10) Dificuldades encontradas: No trabalho de Mello e Xexeo(17), a quantidade de memória utilizada não foi suficiente para treinar a rede neural (*Multilayer Perceptron*) com vetores maiores que 12 bits. Ressalta-se apenas que no trabalho de Mello e Xexeo(20), para realizar as tarefas de classificação com vetores de até 34 bits, foram necessário recursos computacionais melhores, diferindo do perfil de configurações de hardware citadas pelos outros trabalhos. Esta situação deixa um ponto de atenção para a pesquisa atual, pois o processo de treinamento e testes é intensivo de memória e processamento;

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo descrever a fundamentação teórica dos algoritmos criptográficos assimétricos estudados nesta pesquisa, dos algoritmos de aprendizado de máquina e das métricas utilizadas para descrever o desempenho dos algoritmos de classificação. Procura-se subsidiar o leitor desta pesquisa com os fundamentos mínimos para entendimento dos objetos manipulados pelos experimentos (criptogramas) e dos modelos utilizados para atingir o objetivo geral e responder à hipótese desta dissertação.

3.1 Criptografia Assimétrica

A distribuição e o gerenciamento de chaves sempre foi uma atividade crucial nas comunicações criptográficas. Até a década de 1970, as comunicações criptográficas eram realizadas por criptogramas simétricos, ou seja, emissor e destinatário deveriam possuir chaves iguais para realizar a comunicação segura.

Mesmo após a determinação do *Data Encryption Standard* (DES) como primeiro padrão de criptografia, também nesta época, o problema de distribuição de chaves ainda era central para a adoção em escala desta tecnologia. Como garantir que os destinatários recebam a chave de forma segura? À medida em que as interações aumentavam (a Arpanet, precursora da Internet, também se desenvolveu nesta época), os custos com a distribuição de chaves crescia.

Ao se analisar os fatos que levaram ao desenvolvimento de uma solução para os problemas de distribuição e gerenciamento de chaves simétricas, depara-se com pesquisas secretas desenvolvidas pelas agências *UK Government Communication Headquarters* - GCHQ [1] e a *National Security Agency* - NSA (que só vieram a público em 1997 (26)) e pesquisas "públicas" desenvolvidas pelos autores citados a seguir.

Whitfield Diffie e Martin Hellman, influenciados pelo trabalho de Ralph Merkle, divulgaram um método de troca de chaves (27). Esse método utiliza a função alçapão exponenciação em um corpo finito para a transferência de uma chave simétrica em um canal inseguro. Esse foi o primeiro método prático publicado para estabelecer uma chave secreta compartilhada sobre um canal de comunicações inseguro sem o uso de um segredo compartilhado previamente.

Com o aumento exponencial da necessidade de troca de mensagens criptografadas, gerenciar uma grande quantidade de chaves é outro gargalo igualmente custoso à distribui-

[1] Estas pesquisas foram conduzidas por James Ellis, Clifford Cocks e Malcolm Williamson. https://www.gchq.gov.uk/person/malcolm-williamson

ção. No artigo *New Directions in Cryptography* (28), Whitfield Diffie e Martin Hellman apresentaram o conceito de "criptografia de chave pública", onde cada parte envolvida na comunicação possui um par de chaves. Este par é composto por uma chave pública, que deve ser acessível a todos, e uma chave privada, que deverá ser guardada pelo titular.

Este sistema criptográfico realiza duas funções: autenticação/não-repúdio, em que é possível verificar que um determinado emissor, portador de uma chave privada, realmente enviou a mensagem, através da decriptação da mensagem com a chave pública correspondente, e encriptação, em que apenas o portador da chave privada pode decifrar a mensagem encriptada com a chave pública (1). Ambas as funções são apresentadas na Figura 3.

O funcionamento e a segurança da criptografia assimétrica são provenientes de funções alçapão (*trapdoor functions*), onde é fácil computar o resultado da função em um sentido e é difícil de computar na direção oposta, a menos que se tenha alguma informação especial, chamada de alçapão. Jevons (29) [2] já descrevia a fatoração prima de grandes números como um exemplo de uma função alçapão:

> *Given any two numbers, we may by a simple and infallible process obtain their product ; but when a large number is given it is quite another matter to determine its factors. Can the reader say what two numbers multiplied together will produce the number 8,616,460,799? I think it unlikely that anyone but myself will ever know; for they are two large prime numbers, and can only be rediscovered by trying in succession a long series of prime divisors until the right one be fallen upon. The work would probably occupy a good computer for many weeks, but it did not occupy me many minutes to multiply the two factors together.*

Schneier(5) ressalta que desde a década de 1970, diversos algoritmos assimétricos foram propostos. Muitos são inseguros, outros, mesmo considerados seguros, possuem chaves muito grandes ou o texto cifrado é muito maior do que o texto em claro, não sendo atrativos para implementação. Há ainda alguns que são indicados apenas para o processo de assinatura digital e outros para cifrar mensagens. Segundo ele, como já citado no capítulo anterior, apenas três algoritmos são eficientes tanto para o processo de encriptar mensagens quanto para assinar digitalmente: o RSA, ElGamal e Rabin.

Como mencionado no Capítulo 2, já há estudos que abordaram o ataque de distinção em cifras RSA e Curvas Elípticas(vide tabela 3. Portanto, este trabalho também realizou a identificação de padrões utilizando tais algoritmos, para posterior comparação de

[2] Disponível em https://archive.org/details/principlesofscie00jevoiala/page/n8/mode/2up - página 122-123

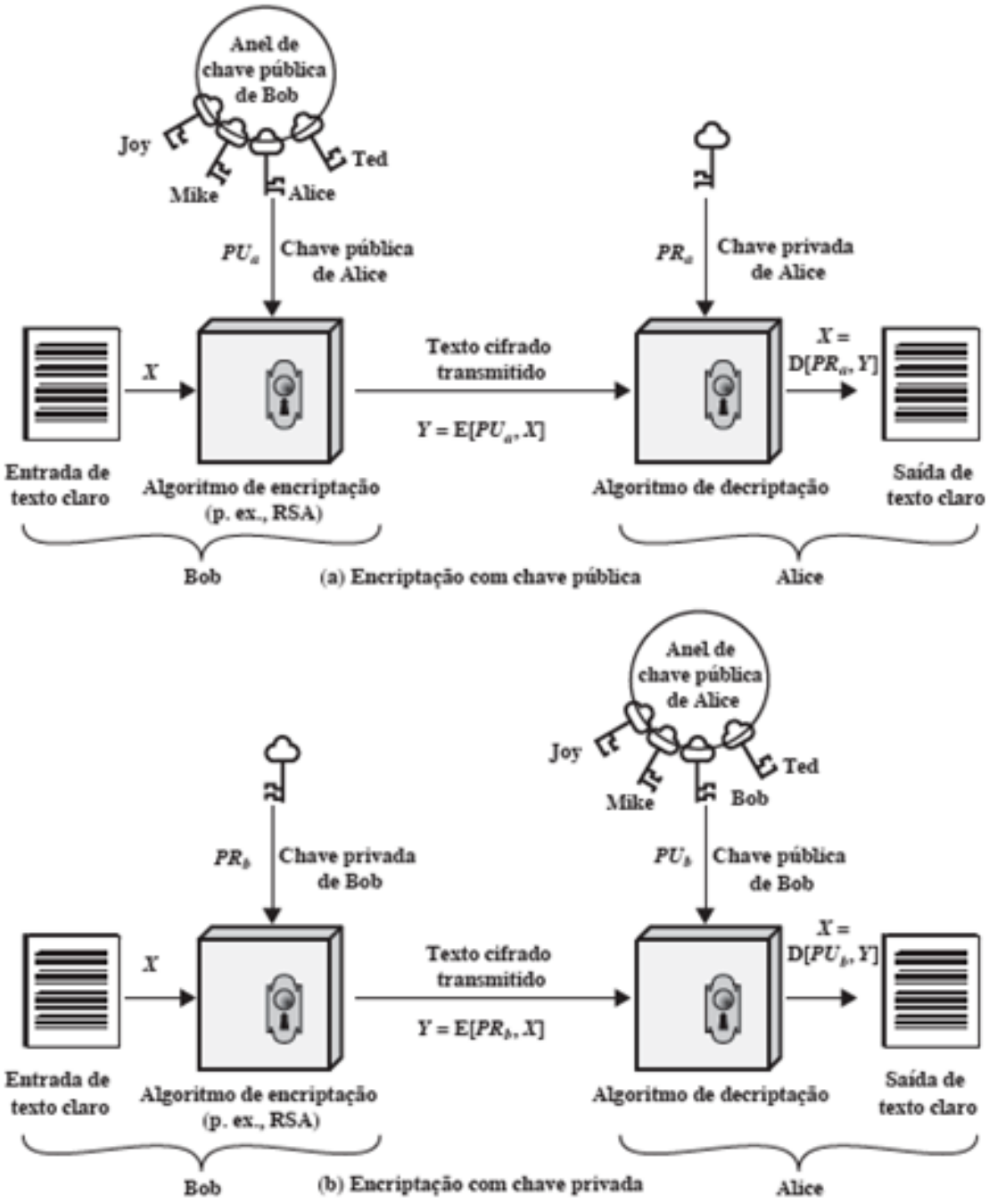


Figura 3 – Criptografia Assimétrica. Fonte: Stallings (1)

resultados, e avaliou os resultados para o algoritmo ElGamal. A seguir, serão apresentados esses algoritmos criptográficos.

3.1.1 O Algoritmo RSA

Influenciados por Diffie e Hellman(28), Ron Rivest, Adi Shamir e Leonard Adleman desenvolveram o algoritmo criptográfico RSA (batizado pelas suas iniciais). Utilizando também exponenciação em aritmética modular, assim como o protocolo de troca de chaves,

foi publicado em 1978 o trabalho intitulado *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems* (30).

Os detalhes do algoritmo são apresentados nas subseções a seguir.

3.1.1.1 Geração das chaves - RSA

O processo de geração do par de chaves assimétricas é descrito no algoritmo a seguir:

Resultado: Geração das chaves pública e privada no RSA

1 Escolha 2 números primos p e q, distintos entre si, e suficientemente grandes;
2 Calcule $n = p.q$;
3 Calcule a função totiente de Euler: $\phi(n) = (p-1)(q-1)$;
4 Escolha um número e, tal que $\text{mdc}(e, \phi(n)) = 1$ e $1 < e < n$;
5 Calcule o inverso multiplicativo de e: $e.d \equiv 1 \pmod{\phi(n)}$;
6 A chave pública é (n, e) e a chave privada é (n, d)

Algoritmo 1: Cálculo das chaves assimétricas no RSA

Atualmente, a forma de utilização do algoritmo é regulada pelos padrões de criptografia de chave pública. Os padrões (*Public Key Cryptography Standards* - PKCS) emitidos pela empresa detentora da patente do RSA, RSA Security LLC, são replicados para a documentação padrão da internet (RFC)[3], gerenciado pela *Internet Engineering Task Force* - IETF.

Como o objetivo deste trabalho é delimitado pelas cifras assimétricas, o método de encriptação adotado é a utilização do algoritmo em seu estado teórico. Sabe-se que, por razões de performance e segurança, a utilização do RSA no mundo real possui uma série de adaptações e complementos, conforme descrito nos padrões de criptografia destacados acima. Normalmente, o algoritmo RSA é utilizado para encriptar a chave de criptografia de um algoritmo simétrico, como por exemplo o AES, e a mensagem em si seria encriptada pelo algoritmo AES. Se fosse adotado esse modelo, este trabalho analisaria criptogramas simétricos, e assim fugiria das delimitações adotadas. Em virtude disso, neste trabalho é utilizada biblioteca de criptografia Pycrypto que implementa o algoritmo RSA em seu "estado teórico", exemplificado a seguir.

3.1.1.2 Cifrar e Decifrar uma mensagem - RSA

O primeiro passo para se começar a cifrar uma mensagem pelo sistema RSA é transformar a mensagem em um número, e isso é feito através do padrão ASCII. Por exemplo, o texto "Instituto Militar de Engenharia" seria transformado em:

73 110 115 116 105 116 117 116 111 (Instituto)

[3] https://www.ietf.org/standards/

77 105 108 105 116 97 114 (Militar)
100 101 (de)
69 110 103 101 110 104 97 114 105 97 (Engenharia)

Em segundo lugar, deve-se quebrar a mensagem em segmentos relativamente pequenos, escolhidos aleatoriamente, para que não seja permitida a técnica de análise de frequência na tentativa de quebra do código. Estes segmentos devem ser menores que o tamanho da chave privada.

Suponhamos que a mensagem, transformada em uma sequência numérica, seja dividida em k segmentos S.

$$\text{m} = \quad S_1 \quad S_2 \quad S_3 \quad S_{...} \quad S_k$$

Para enviar uma mensagem a um destinatário, primeiramente deve-se ter acesso à sua chave pública (n, e). Para cifrar a mensagem, deve-se calcular cada segmento S_i, com $1 < i < k$:

$$S_i^e \pmod{\text{n}} \equiv C_i$$

Então o remetente envia para o destinatário portador da chave privada a mensagem *m'* composta pela sequência de C_i.

$$\text{m'} = \quad C_1 \quad C_2 \quad C_3 \quad C_{...} \quad C_k$$

Para decifrar a mensagem, o portador da chave privada (n, d) deve calcular, para cada segmento, a equação abaixo, chegando à mensagem original:

$$C_i^d \pmod{\text{n}} \equiv S_i$$

3.1.2 O Algoritmo ElGamal

Como citado na introdução deste capítulo, o algoritmo ElGamal (31) tem sua segurança na dificuldade de cálculo de logaritmos discretos em um corpo finito (mesma função alçapão do protocolo Diffie-Hellman). É considerada uma tarefa computacionalmente intratável encontrar x, dados a, b e n inteiros:

$$a^x \equiv b \pmod{\text{n}}$$

3.1.2.1 Geração das chaves - ElGamal

O algoritmo de geração de chaves no ElGamal funciona do seguinte modo:

Resultado: Geração das chaves pública e privada no ElGamal
1 Escolha um número primo p e dois números aleatórios g e x, tal que g e $x < p$;
2 Calcule y = g^x (mod p);
3 A chave pública é (p, g, y) e a chave privada é (x);

Algoritmo 2: Cálculo das chaves assimétricas no ElGamal

3.1.2.2 Cifrar e decifrar uma mensagem - ElGamal

O processo de cifrar e decifrar uma mensagem é semelhante ao apresentado na seção anterior (RSA).

Para cifrar uma mensagem M, primeiramente escolha um número aleatório k, tal que k é um primo relativo de $p - 1$. Então calcule:

Resultado: Mensagem cifrada e decifrada
1 $a = g^k$ (mod p);
2 $b = y^k.M$ (mod p);
3 O par (a,b) é a mensagem cifrada;
4 Calcule $M = b/a^x$ (mod p), utilizando a chave secreta x para decifrar a mensagem;

Algoritmo 3: Cifrar e decifrar uma mensagem no ElGamal

Note que, em comparação com o RSA, a mensagem cifrada (a,b) é o dobro do tamanho.

3.1.3 Curvas elípticas

Posteriormente, nos anos 80, Koblitz(32) e Miller(33) propuseram, independentemente, criptossistemas de chaves públicas baseados no problema do logaritmo discreto baseado em grupos formados por pontos em uma curvas elípticas, em vez do grupo multiplicativo em corpo finito.

Pode-se adaptar os algoritmos criptográficos ao contexto de curvas elípticas em corpos finitos. Cita-se, por exemplo, Demytko(34) que propôs uma adaptação do algoritmo RSA sobre curvas elípticas e Sutikno, Surya e Effendi(35) que utilizaram o algoritmo ElGamal.

Para este trabalho, será utilizado o problema do logaritmo discreto em um grupo de pontos de curva elíptica. Analogamente ao cálculo de $y = g^x(modp)$ no ElGamal, calcular $Q = lP$, em que P e Q são pontos de uma curva elíptica e l é um número natural, é possível

computacionalmente quando se possui P e l. Agora, dado P e Q, é computacionalmente intratável calcular-se l.

O documento NIST SP 800-186 (36), recomenda as curvas elípticas P-224, P-256, P-384, P-251, Curva448 e Curva25519 para o uso em criptografia baseada em problemas do logarítmo discreto. Para esta pesquisa, foi adotada a curva P-256.

3.1.3.1 Cifrar e Decifrar uma mensagem - CURVAS ELÍPTICAS

O algoritmo para cifrar e decifrar uma mensagem em curvas elípticas é apresentado a seguir. A transformação da mensagem em texto para números pode ser realizada da mesma forma apresentada anteriormente.

Resultado: Mensagem cifrada e decifrada

1 O emissor e destinatário devem estabelecer previamente a curva elíptica e um ponto P;
2 O destinatário escolhe um número $b \in N^*$ como chave privada;
3 Envia ao emissor $K = bP$ (chave pública);
4 O emissor mapeia o alfabeto em pontos da curva elíptica;
5 O emissor deve transformar a mensagem m a ser enviada em pontos da curva elíptica, conforme mapeamento no passo anterior ($m' = mP$);
6 O emissor escolhe outro número $a \in N^*$ e calcula $c_1 = aP$ e $c_2 = m' + aK$;
7 Para decifrar a mensagem m, o destinatário calcula $c_2 - bc_1$ e descobre m'. A partir do mapeamento inicial dos pontos, chega-se a m;

Algoritmo 4: Cifrar e decifrar uma mensagem em Curvas Elípticas

3.2 Algoritmos de Aprendizado de Máquina

A inteligência artificial (IA) é um ramo de pesquisa da ciência da computação que busca construir mecanismos e/ou dispositivos que simulem a capacidade do ser humano de pensar e resolver problemas (2). Com a crescente complexidade dos problemas a serem tratados e do volume de dados gerados, a necessidade de ferramentas computacionais mais autônomas tornou-se um imperativo. Tais ferramentas precisam ter a habilidade de criar uma hipótese, ou função, a partir das experiências passadas, capaz de resolver o problema que se queira tratar. A esse processo de indução de uma hipótese chama-se Aprendizado de Máquina (AM).

O aprendizado indutivo é efetuado a partir de raciocínio sobre exemplos fornecidos por um processo externo ao sistema de aprendizado. O aprendizado indutivo pode ser dividido em supervisionado e não-supervisionado. No aprendizado supervisionado é fornecido ao algoritmo de aprendizado um conjunto de exemplos de treinamento para os

quais o rótulo da classe associada é conhecido. Já no aprendizado não-supervisionado, o algoritmo analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou *clusters*. Após a determinação dos agrupamentos, normalmente, é necessária uma análise para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado (37).

Além dessa classificação binária de supervisionado e não supervisionado, Gama et al.(2) classificam os métodos de AM sob outro aspecto de como o aprendizado acontece:

1. Modelos baseados em distâncias: Uma forma de classificar um exemplo é lembrar de outro similar cuja classe é conhecida e supor que o novo exemplo terá a mesma classe. A hipótese central é que os exemplos similares estão concentrados em uma mesma região no espaço de entrada.

2. Modelos Probabilísticos: A ideia geral consiste em utilizar modelos estatísticos para encontrar uma boa aproximação do conceito induzido. Dentre os métodos estatísticos, destacam-se os de aprendizado Bayesiano, que utilizam um modelo probabilístico baseado no conhecimento prévio do problema, o qual é combinado com os exemplos de treinamento para determinar a probabilidade final de uma hipótese.

3. Modelos baseados em Procura: O problema de aprendizado de máquina pode ser interpretado como um problema de procura em um espaço de possíveis soluções. Dada uma linguagem para representar generalizações dos exemplos e uma função de avaliação de hipóteses, o algoritmo procura no espaço de hipóteses a de melhor resultado.

4. Modelos baseados em Otimização: Nesses tipos de modelos, o aprendizado é representado pela otimização de alguma função, podendo maximizar uma variável resultado ou minimizar a taxa de erro.

Há também a possibilidade de combinar diferentes tipos de classificadores, objetivando obter melhores resultados de classificação. Um classificador *Ensemble* consiste em um meta-algoritmo formado por um conjunto de classificadores treinados individualmente cujas decisões são de alguma forma combinadas. Podem ser divididos em:

1. *Bagging*: sua denonimação vem de *bootstrap aggregating* e geralmente considera algoritmos de AM homogêneos, o aprendizado é independente um do outro (em paralelo) e os resultados são combinados seguindo algum tipo de média. Os dados de treinamento para cada classificador são gerados aleatoriamente a partir do dataset inicial (técnica *boostrap*). A amostragem *boostrap* é uma técnica de amostragem com resposição. A partir do conjunto de treinamento inicial, são selecionados aleatoriamente exemplos para um novo subconjunto de treinamento.

2. *Boosting*: Este método funciona da mesma maneira que os métodos de *bagging*. É construída uma família de algoritmos que são agregados para obter um classificador forte e com melhor desempenho. No entanto, diferentemente do *bagging*, que visa principalmente reduzir a variância, o *boosting* é uma técnica que consiste em ajustar sequencialmente vários algoritmos fracos de uma maneira adaptativa: cada modelo na sequência é ajustado, dando mais importância às observações no conjunto de dados que foram rotuladas incorretamente pelo classificador anterior. Intuitivamente, cada novo modelo concentra seus esforços nos erros dos anteriores, para que, no final do processo, se tenha um classificador forte com viés mais baixo.

3. *Stacking*: Neste método, classificadores de diferentes tipos são combinados (por exemplo, classificadores baseados em distância, redes neurais e árvores de decisão). O dataset inicial é dividido em dois. No primeiro subconjunto de dados, os classificadores são treinados e suas predições são *input* de um outro classificador, que será testado com a segunda metade do dataset.

Os classificadores empregados nesta pesquisa são o *Support Vector Machines* (SVM), *Random Forest* e *K Nearest Neighbor* (KNN). Esta escolha está associada, conforme descita no capítulo 2, à lacuna de trabalhos que analisem cifras assimétricas através destes algoritmos.

Estes três algoritmos abordam a tarefa de classificação de maneiras distintas, de acordo com os modelos apresentados anteriormente. Como será apresentado nas próximas seções, o KNN utiliza o modelo baseado em distâncias para executar o aprendizado e a predição; o SVM utiliza o modelo baseado em procura de alguma superfície de decisão de classificação e o *Random Forest* é um classificador do tipo *Ensemble* que também utiliza um modelo baseado em procura de superfícies de decisão (árvores de decisão).

3.2.1 *K Nearest Neighbor*

O algoritmo K Nearest Neighbor (KNN) classifica um novo objeto com base nos K exemplos do conjunto de treinamento que são mais próximos a ele, baseado no paradigma de aprendizado indutivo *"Objetos com características semelhantes pertencem ao mesmo grupo"*. É classificado como um algoritmo "preguiçoso", porque toda a computação é adiada até a fase de classificação, já que o processo de aprendizado consiste apenas em memorizar os objetos.

Cada objeto representa um ponto em um espaço definido pelos atributos, denominado espaço de entrada. Definindo uma métrica nesse espaço, é possível calcular as distâncias entre cada dois pontos. A métrica mais utilizada é a distância euclidiana dada pela equação:

$$d(x_i, y_i) = \sqrt{\sum_{l=1}^{d}(x_i^l - x_j^l)^2} \tag{3.1}$$

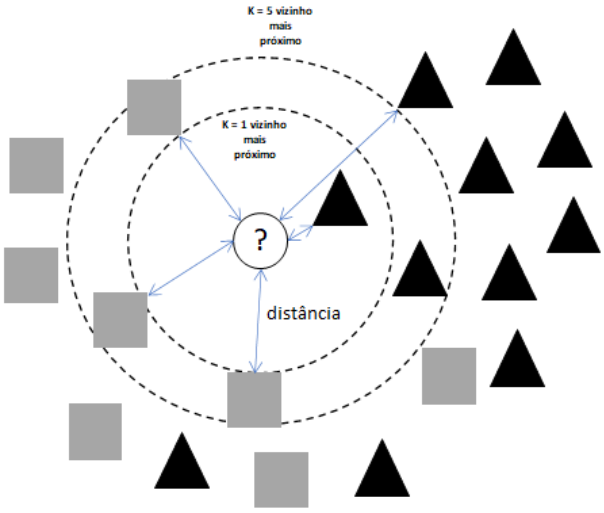


Figura 4 – KNN - fonte: Adaptado de (2)

Na Figura 4, se considerarmos $K = 1$, a classificação do novo objeto será a classificação triângulo. Se utilizarmos $K = 5$, a classificação será quadrado. A escolha do valor de K mais apropriado para um problema de decisão específico pode não ser trivial. Esta variável, definida pelo usuário, frequentemente é pequena e ímpar, para não haver problemas de empate.

O gráfico representado na Figura 5 é uma representação em 2D, através de uma redução de dimensionalidade utilizando a técnica PCA, dos dados utilizados para treinamento e teste. Os pontos em laranja são os pertencentes ao grupo ElGamal, azul são RSA e verde são Curvas Elípticas. Como é possível verificar, no centro do gráfico, as 3 classes se misturam. Portanto, o algoritmo KNN ao classificar um novo exemplo, terá menores chances de acerto. No entanto, se a nova mostra estiver nas bordas, por exemplo um ponto "?1" = (0.001, 0.009), onde os pontos em verde são mais próximos, o algoritmo classificará como Curvas Elípticas, e terá maior probabilidade de acerto, pois nenhuma outra classificação possui exemplos naquela região. Já o ponto "?2" = (0.005, 0.002) tem como pontos próximos as classificações RSA e Curvas Elípticas. Como os pontos azuis estão mais próximos, o algoritmo retornaria à classificação do ponto "?2" como RSA.

3.2.2 *Random Forest*

Random Forest (RF) é um classificador *ensemble* caracterizado como uma generalização do classificador Árvore de Decisão. Proposto por Breiman(38), é definido como uma coleção de classificadores Árvore de Decisão $\{h(X, \Theta_k), k = 1, \ldots\}$, onde Θ_k são vetores

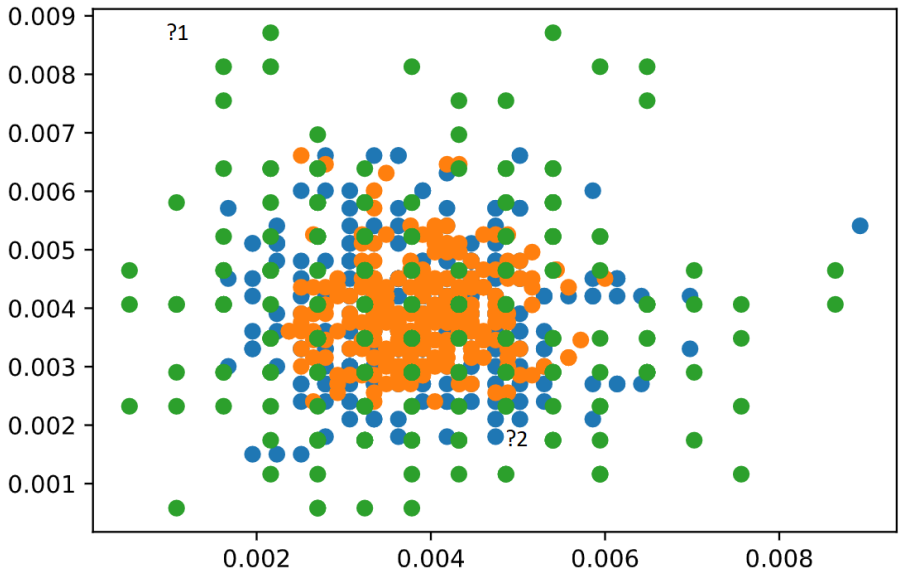


Figura 5 – Visualização em 2D dos dados de treinamento e teste - Fonte: O autor.

aleatórios independentes e idênticos compostos por um subconjunto dos atributos, e cada árvore determina um voto unitário para a classe mais popular do vetor de entrada X.

Dado um conjunto de classificadores $h_1(X), h_2(X), \ldots, h_k(X)$ treinados por vetores aleatórios Θ_k, definimos a função de margem (mg):

$$mg(X,Y) = avg_k(h(X) = Y) - max_{j \neq Y} avg_k(h_k(X) = j) \tag{3.2}$$

Esta função define a média do número de votos das árvores que votaram na classe Y, presente no vetor de classificações possíveis X, menos o valor máximo de votos em uma classificação diferente de Y. A margem mede até que ponto o número médio de votos em Y para a classe certa excede a média de votos para qualquer outra classe. Quanto maior a margem, mais confiança na classificação.

Durante a fase de treinamento, o RF produz amostras aleatórias de conjuntos de treinamento (amostras *bootstraps*) para cada árvore, conforme o método *bagging*. Breiman(38) justifica o uso do método *bagging* por duas razões: reduzir a complexidade dos modelos, de forma a evitar a ocorrência de superajuste dos dados por modelos muito complexos e a possibilidade de ser utilizado para fornecer estimativas contínuas do erro de generalização do conjunto combinado de árvores, assim como estimativas para força e correlação.

Ao final o algoritmo constrói sua decisão por meio da contagem dos votos das árvores em cada classe e, em seguida, seleciona a classe vencedora em termos de número de votos acumulados dentre todas as árvores.

No contexto desta pesquisa, os dados de treinamento formam as diversas árvores da floresta. Os dados de teste e validação são classificados mediante estas florestas formadas com os dados de treinamento.

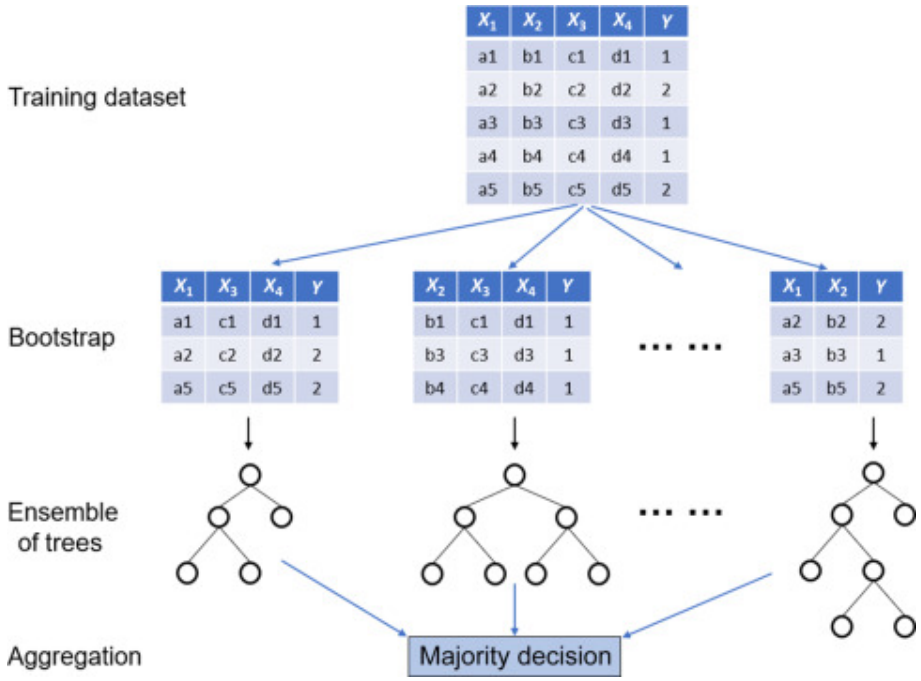


Figura 6 – Random Forest - fonte: Science Direct

No contexto desta pesquisa, durante o processo da formação das amostras *bootstrap*, são selecionados aleatoriamente subconjuntos compostos por um subgrupo dos 256 atributos (posições do vetor) de cada elemento, por exemplo: Amostra 1 - atributos 1, 4, 45, 78, 132, 201; Amostra 2 - atributos 3, 62, 94.

Para cada cenário de amostras, são construídas árvores de decisão, para a classificação dos elementos de cada grupo. Ao classificar uma determinada amostra, esta é avaliada em cada cenário e é realizada uma decisão majoritária, combinando os resultados das árvores de decisão.

3.2.3 *Support Vector Machines*

O objetivo do algoritmo de classificação *Support Vector Machines* (SVM) é encontrar um hiperplano em um espaço N-dimensional que consiga separar os vetores de acordo com a sua classificação. Este modelo é embasado pela Teoria de Aprendizado Estatístico (TAE). Esta teoria estabelece condições matemáticas que auxiliam na escolha de um classificador particular a partir de um conjunto de dados de treinamento. (2)

Na TAE, assume-se inicialmente que os dados do domínio em que o aprendizado está ocorrendo são gerados de forma independente e identicamente distribuída de acordo

com uma distribuição de probabilidades $P(x, y)$ que descreve a relação entre os objetos e seus rótulos. (39)

A distância de separação criada pelo hiperplano de classificação e as amostras de treinamento é chamado de erro ou risco. O classificador SVM busca obter um equilíbrio entre o erro com relação ao conjunto de treinamento (risco empírico) e o erro com relação do conjunto de testes (risco de generalização), minimizando o excesso de ajustes (*overfitting*) que podem reduzir a capacidade de generalização do classificador.

A questão da generalização pode ser mais bem compreendida para um caso de duas classes. Assumindo que o conjunto de treinamento para duas classes são linearmente separáveis, a função de decisão mais adequada é aquela que maximiza a distância entre as amostras (maximização da margem). Essa abordagem minimiza a probabilidade de erro de generalização. A Figura 7 ilustra este exemplo. Existem diversos hiperplanos que podem separar as amostras das duas classes (gráfico da esquerda). Porém cada hiperplano possui uma chance de erro de separar corretamente novas amostras. O hiperplano que determina a distância máxima entre os pontos das duas classes é aquele que minimiza o erro de generalização.

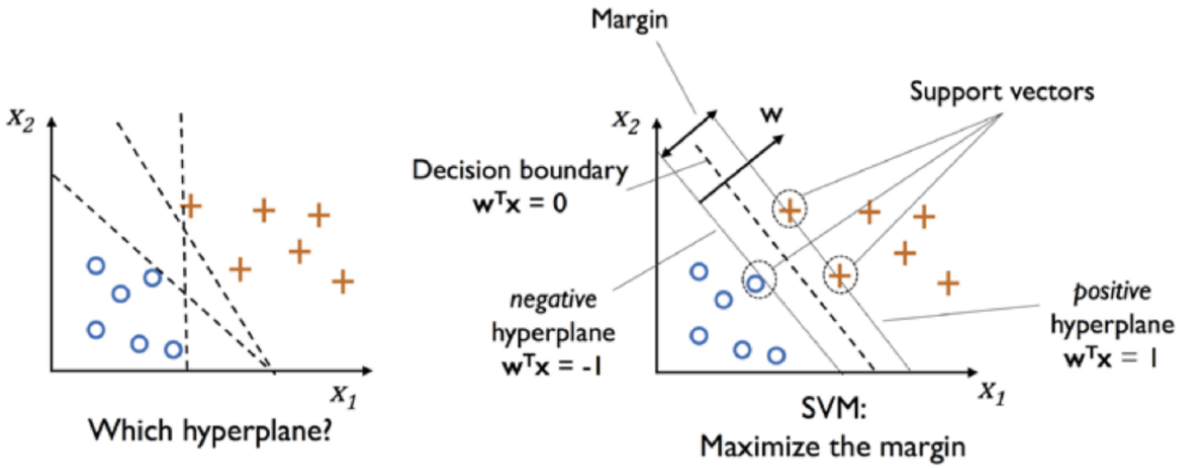


Figura 7 – Hiperplanos lineares de separação - Fonte: Raschka e Mirjalili

O cenário de classificação mais simples é quando as amostras das classes puderem ser separadas linearmente. Então, seja X um conjunto de treinamento com n objetos $x_i \in X$ e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço de entrada e $Y = \{-1, +1\}$ são as possíveis classes.

A presença de ruídos e outliers nos objetos de treinamento podem dificultar a identificação dos hiperplanos de separação. Desta forma, permite-se que alguns objetos possam violar a restrição e estarem dentro da margem de separação. Isso é feito com a introdução de variáveis de folga ϵ_i:

$$Minimizar\ \frac{1}{2}||w||^2 + C\Big(\sum_{i=1}^{n}\epsilon_i\Big) \tag{3.3}$$

A constante C é um termo de regularização que impõe um peso à minimização dos erros no conjunto de treinamento. Este parâmetro equilibra dois objetivos diferentes: maximizar a margem e minimizar o número de erros nos dados de treinamento.

Quando C é muito pequeno, a soma dos termos de erro torna-se desprezível na função objetivo (3.3), de forma que o objetivo da otimização é maximizar a margem. Como resultado, a margem pode ser grande o suficiente para conter todos os pontos. Em outro extremo, quando C é muito grande, a soma dos termos de erro domina o termo margem na função objetivo, de forma que o objetivo da otimização é minimizar a soma dos termos de erro. Como resultado, a margem pode ser tão pequena que não contém pontos. Observe que, apesar das diferenças no valor do parâmetro C e no tamanho da margem, todos os quatro hiperplanos mostrados na Figura 8 são bastante semelhantes e todos classificam corretamente os mesmos pontos de treinamento.

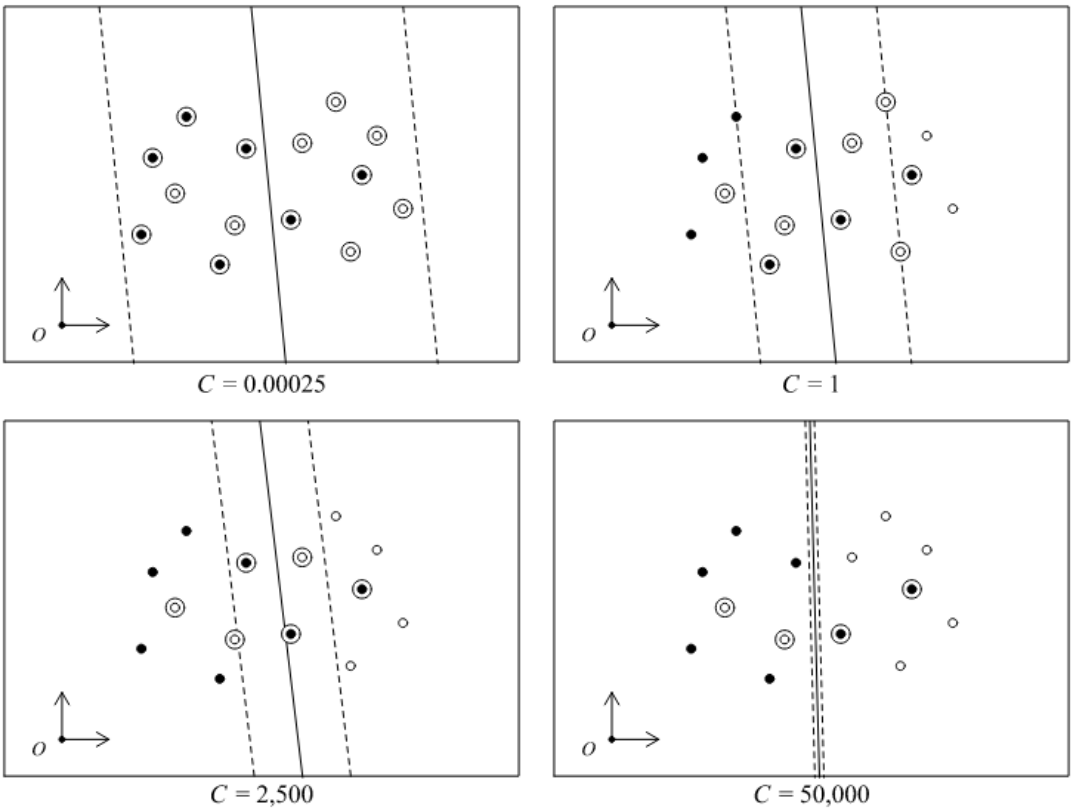


Figura 8 – Explicação da variação do C. Fonte: site svmtutotial.online

No entanto, pode-se encontrar exemplos onde o conjunto de dados não é linearmente separável, mesmo com a presença de ruídos e *outliers*. As SVMs lidam com problemas não lineares mapeando o conjunto de treinamento de seu espaço original para um novo espaço

de maior dimensão, utilizando uma função $\Phi : X \rightarrow \Omega$. A escolha apropriada de Φ faz com que o conjunto de treinamento mapeado em Ω possa ser separado por uma SVM Linear.

Esta transformação é motivada pelo Teorema de Cover. Dado um conjunto de dados não linear no espaço de entradas X, esse teorema afirma que X pode ser transformado em um espaço de características Ω no qual, com alta probabilidade, os objetos são linearmente separáveis. Para isso, duas condições devem ser satisfeitas. A primeira é que a transformação seja não linear, enquanto a segunda é que a dimensão do espaço Ω seja suficientemente alta.

Deste modo, a computação de Φ pode ser inviável, devido à alta dimensionalidade. No entanto, independentemente da dimensão, a única informação necessária sobre o mapeamento é a realização do cálculo de produtos escalares entre os dados transformados. Para resolver esse problema, o SVM faz uso de funções denominadas *Kernel*. Uma função *Kernel* (K) recebe dois pontos do conjunto de entrada e computa o produto escalar no espaço Ω.

As principais funções *Kernel* são:

Tipo de Kernel	Função $K(x_i, x_j)$	Parâmetros
Polinomial	$(\delta(x_i.x_j) + \kappa)^d$	δ, κ, d
Radial Basis Function (RBF)	$\exp(-\gamma\|\|x_i - x_j\|\|^2)$	γ
Sigmoidal	$\tanh(\delta(x_i.x_j) + \kappa)$	δ, κ

Tabela 4 – Funções Kernel mais comuns. Fonte: Gama et al.(22)

A obtenção de um classificador por meio do uso de SVMs envolve então a escolha de uma função kernel, além de parâmetros dessa função e do valor da constante de regularização C, pois eles definem a fronteira de decisão. (2)

3.3 Medidas de Avaliação

Para mensurar o desempenho dos algoritmos de classificação, são necessários métodos e métricas de avaliação. Os métodos de avaliação determinam como a base de dados será dividida de forma que a etapa de treinamento não seja feita com os mesmos dados que a etapa de teste, onde busca-se avaliar a capacidade de generalização do classificador para determinar o tipo de dado, frente a um dado nunca antes analisado. Já as métricas de avaliação são valores numéricos quantitativos utilizados para mensurar o desempenho do algoritmo frente a algum critério.

Para o cenário desta pesquisa, em que a quantidade de exemplos para treinamento e classificação é grande, o método de validação cruzada *Holdout* foi possível. Neste método, a base de dados foi separada em bases disjuntas de treinamento, teste e validação. As

amostras de teste foram utilizadas para o cálculo das medidas de desempenho e as amostras de validação foram utilizadas para legitimar os resultados alcançados na fase de testes.

As métricas que foram utilizadas neste estudo são: acurácia, precisão, *recall* e *f1score*. Estas medidas foram apontadas em Sokolova e Lapalme(40) como as principais medidas de desempenho em algoritmos de classificação multiclasse.

Para a construção destas medidas, foi utilizada a matriz de confusão, que é uma tabela comparativa entre as predições do algoritmo de classificação e as reais classificações. De forma conceitual, para um classificador de apenas uma classe, tem-se a matriz:

Matriz de Confusão	Prev Classe	Prev Não Classe
Real Classe	Positivo Verdadeiro (PV)	Falso Negativo (FN)
Real Não Classe	Falso Positivo (FP)	Negativo Verdadeiro (NV)

Tabela 5 – Modelo de matriz de confusão

A seguir, serão definidas as fórmulas de cada medida:

- Acurácia ACC é a medida da efetividade do classificador. Ela descreve que o algoritmo consegue predizer tanto que uma amostra é de uma determinada classe (PV), quanto dizer que uma outra amostra não faz parte de uma determinada classe (NV).

$$ACC = \frac{PV + NV}{PV + FN + FP + NV} \times 100\% \tag{3.4}$$

- A Precisão é definida pela divisão do número de amostras classificadas corretamente pelo algoritmo pelo total de amostras classificadas da classe. Para uma predição com apenas uma classe, uma precisão de 50% equivale a uma predição aleatória. A quantidade de positivos verdadeiros é igual a quantidade de falsos positivos.

$$Precisão = \frac{PV}{PV + FP} \times 100\% \tag{3.5}$$

- *Recall* é a medida do classificador para identificar corretamente todos os casos da classe. O *Recall* significa quantos que são realmente da classe alvo foram rotulados corretamente.

$$Recall = \frac{PV}{PV + FN} \times 100\% \tag{3.6}$$

- F1score é a média harmônica entre Precisão e *Recall*. É comumente utilizada para se avaliar de uma maneira única um classificador de acordo com a classe.

$$F1 = \frac{2.Precisão.Recall}{Precisão + Recall} \times 100\% \tag{3.7}$$

4 EXPERIMENTOS

4.1 Descrição dos Experimentos

Para avaliar a existência de padrões nos criptogramas, foi desenhado o experimento a seguir, composto por 6 etapas:

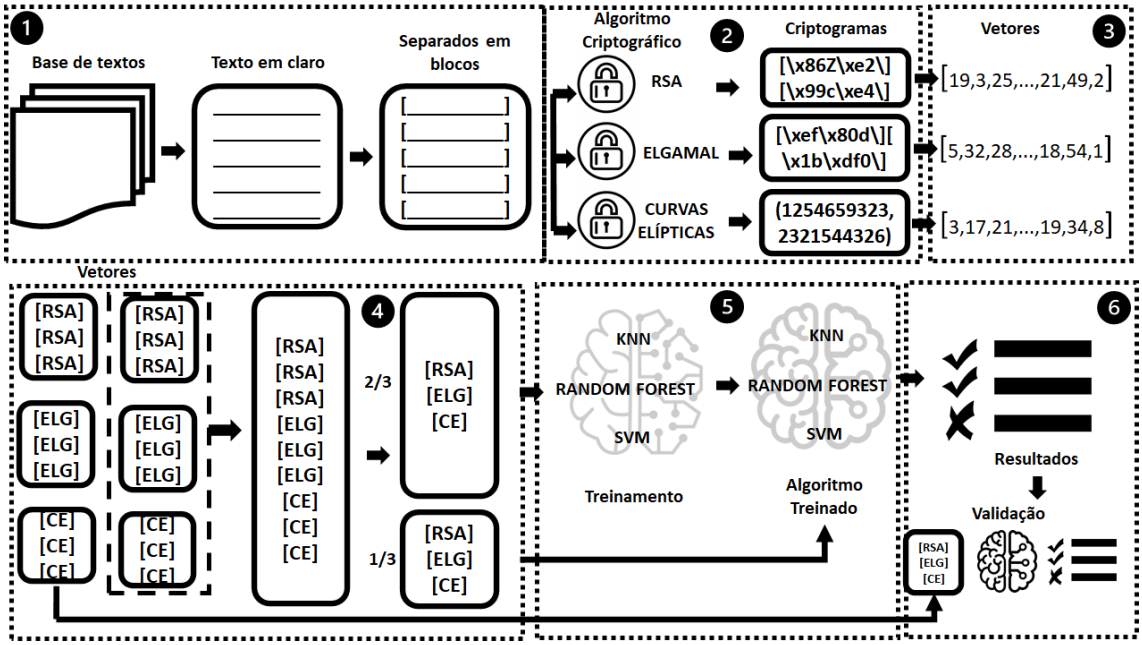


Figura 9 – Processos básicos do experimento: Encriptação (1, 2), Vetorização (3), Treinamento (4 e 5) e Classificação (6)

De maneira geral, cada texto da base de dados foi separado em blocos de 1024 *bits* (Etapa 1). Em seguida, cada bloco foi criptografado e todos os blocos pertencentes ao mesmo texto e de uma mesma cifra foram concatenados formando o criptograma. Este processo foi repetido para os três algoritmos criptográficos - RSA, ElGamal e Curvas Elípticas (Etapa 2). Após a geração dos criptogramas, foi realizado o pré-processamento dos criptogramas, conforme descrito na seção 4.1.3 (Etapa 3). As 3 primeiras etapas foram executadas duas vezes, formando duas bases diferentes para cada algoritmo criptográfico. Uma das bases foi utilizada para treinamento e teste, enquanto a segunda base foi utilizada para validação dos resultados (Etapa 4). Após a separação das bases de treinamento e teste, os classificadores KNN, *Random Forest* e SVM são treinados (Etapa 5), testados e validados (Etapa 6) utilizando as devidas bases de vetores. Mais detalhes sobre cada etapa estão descritos nas subseções a seguir.

4.1.1 Etapa 1 - Base de dados

A seleção da base de dados para a realização desta pesquisa foi baseada em dois critérios principais:

1. Tamanho da base de dados: Para se obter uma boa representatividade de amostras de criptogramas gerados por diversos tamanhos de textos e maximizar a qualidade do treinamento dos classificadores, buscou-se uma base com, no mínimo, a maior quantidade de textos utilizados pelos trabalhos anteriores que estudaram também cifras assimétricas. Sendo assim, nossa referência é o trabalho de Mello e Xexeo(20) que utilizou um base de 4.200 amostras de textos.

2. Disponibilidade: A base de dados ser disponível para outros pesquisadores, a fim de possibilitar replicação dos experimentos e críticas a este trabalho.

Atendendo a ambos os critérios, foi selecionada uma base de textos retirados da Wikipedia no *site* Kaggle [1], utilizada pela comunidade de pesquisadores em aprendizado de máquina, com 30.279 textos de diversos tamanhos, contabilizando mais de 483 milhões de caracteres. Ressalta-se que, por ser oriundo da Wikipedia, os textos não são escritos por uma pessoa apenas, o que diminui a possibilidade de existir viés oriundo do estilo literário do autor. A Figura 10 ilustra o perfil da base escolhida. A quantidade identificada em cada classe significa o número de textos da base que possuem a quantidade de *bits* de acordo com a faixa correspondente.

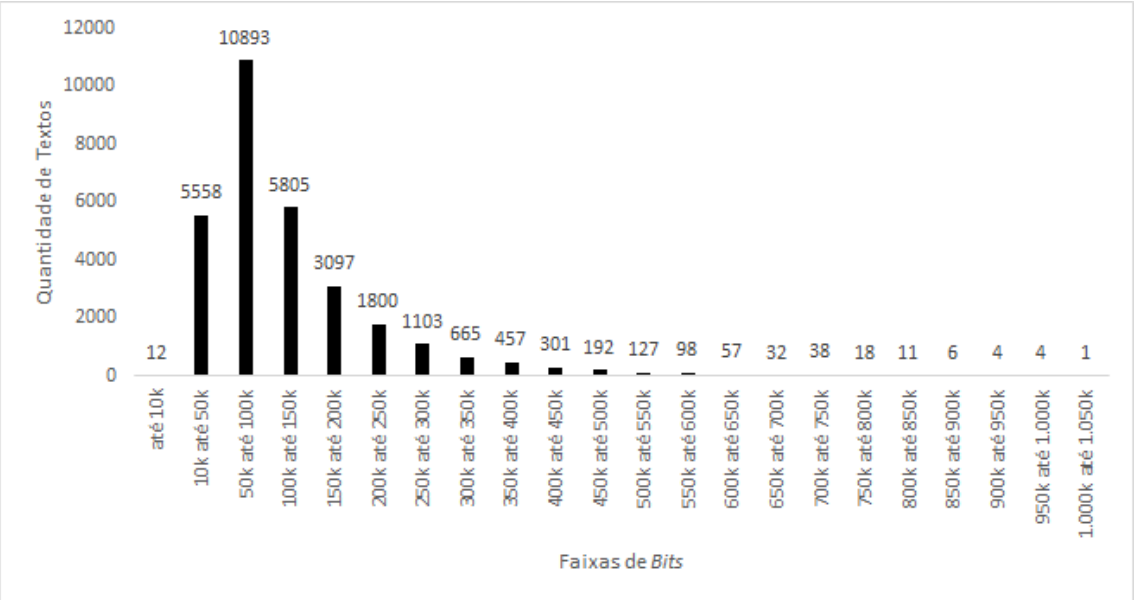


Figura 10 – Perfil do tamanho dos textos da base de dados - em *bits*

[1] https://www.kaggle.com/urbanbricks/wikipedia-promotional-articles

4.1.2 Etapa 2 - Encriptação

Cada bloco de texto em claro foi criptografado com os algoritmos RSA, ElGamal e Curvas Elípticas (CE), conforme parâmetros descritos na Tabela 6. As bibliotecas em python utilizadas e suas justificativas estão descritas no anexo B. Após o processo de encriptação, todos os blocos correspondentes a um mesmo texto e algoritmo criptográfico foram consolidados em um mesmo criptograma, para posterior vetorização.

Algoritmo	Tamanho da chave	Repetição de chaves
RSA	1024 bits	Uma chave por texto
ElGamal	1024 bits	Uma chave a cada 300 textos
CE	160 bits	Uma chave por texto

Tabela 6 – Condições de encriptação

Durante o processo de encriptação, foram geradas novas chaves para cada um dos 30.279 textos da base. Especificamente, para o caso do algoritmo assimétrico ElGamal, foram geradas 100 chaves diferentes para encriptar a base toda. A justificativa para este uso é devido ao tempo superior de geração de novas chaves; enquanto uma chave do algoritmo RSA é gerada em 0,46 segundos, uma chave de ElGamal é gerada em 24,47 segundos. Gerar uma chave ElGamal para cada um dos textos da base consumiria um tempo total de 205 horas apenas nesse passo do algoritmo. Por isso foi adotada a abordagem da geração de 100 chaves para encriptar a base inteira com o algoritmo ElGamal. Ao inspecionar o código utilizado para a geração das chaves, foi identificado que o método de geração de chaves do RSA foi baseado no trabalho de Silverman(41), com um método mais rápido para a geração dos números primos, desta forma justifica o tempo menor de geração das chaves.

Porém, buscou-se manter a diretriz de mudança de chaves a cada mensagem. Para isso, a cada bloco de 1024 *bits* de um mesmo texto, é gerada uma nova "chave de sessão". Conforme descrito na fundamentação teórica 3.1.2, a chave de sessão é um valor aleatório utilizado no algoritmo a cada envio de mensagem. Portanto, para os textos que compartilham a mesma chave base (Número primo P, Gerador G e chave privada X), há várias chaves de sessão utilizadas para encriptar cada texto, conforme o seu tamanho. Entende-se que, desta forma, é minimizado qualquer influência do uso de menos chaves para os criptogramas do ElGamal, em comparação às demais cifras RSA e Curvas Elípticas.

O uso do tamanho de chaves de 160 bits na criptografia de curvas elípticas foi adotado para que os criptogramas utilizados nesta pesquisa possuam o mesmo nível de segurança, conforme (3) e ilustrado na Figura 11.

A Figura 12 a seguir mostra o histograma do tamanho dos criptogramas em *bits* gerados pelos três algoritmos.Os criptogramas foram separados em faixas de cinquenta mil,

	Security level (bits)				
	80 (SKIPJACK)	112 (Triple-DES)	128 (AES-Small)	192 (AES-Medium)	256 (AES-Large)
DL parameter q EC parameter n	160	224	256	384	512
RSA modulus n DL modulus p	1024	2048	3072	8192	15360

Figura 11 – Tamanho das chaves e nível de segurança - Fonte: Hankerson, Menezes e Vanstone (3)

após os primeiros 10 mil *bits*. O somatório do histograma de cada algoritmo totaliza 30.279, correspondendo ao número de textos da base original. O menor tamanho de criptograma é de 4076 *bits*, correspondente ao algoritmo de Curvas Elípticas, e o maior é 2.287.464 *bits*, do ElGamal.

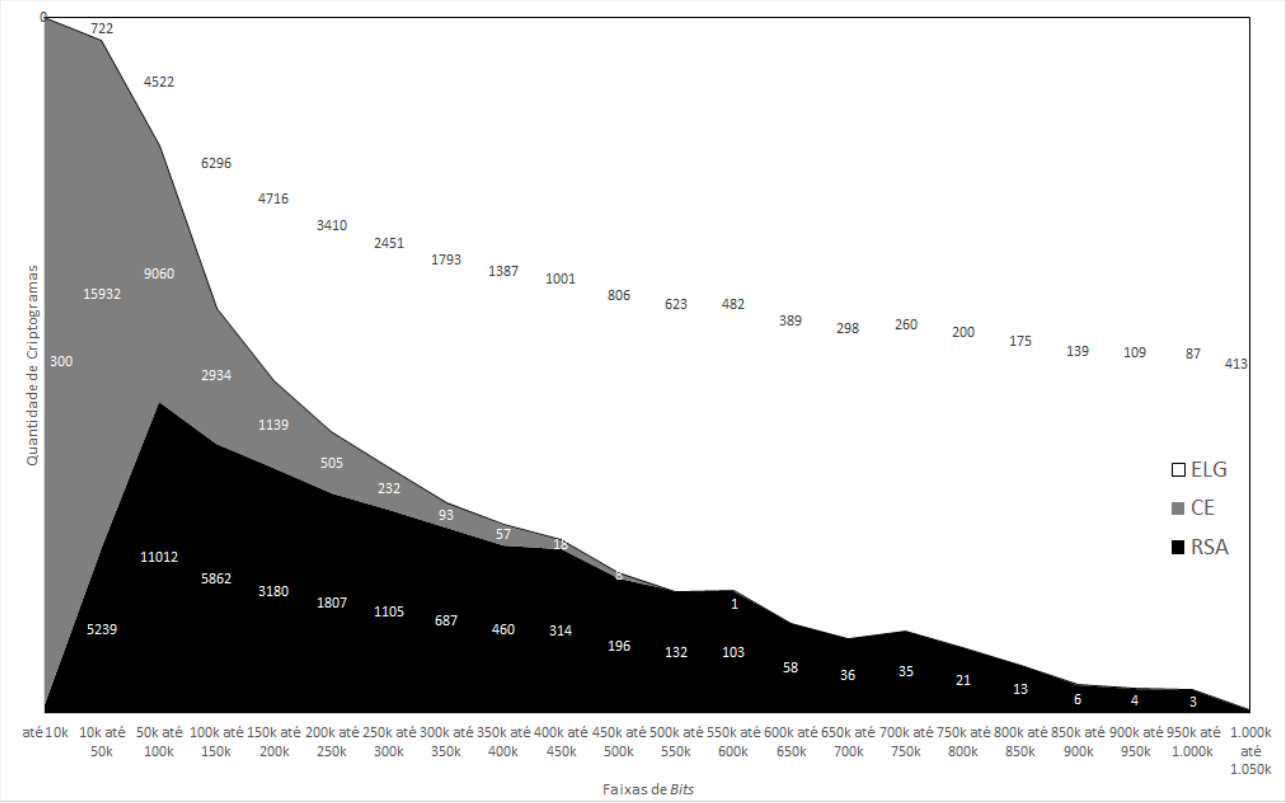


Figura 12 – Distribuição de criptogramas por tamanho e por algoritmo

Pode-se perceber que os criptogramas das Curvas Elípticas (CE) possuem um tamanho menor - em sua maioria localizados até a faixa dos 100 mil *bits*, enquanto os criptogramas do ElGamal possuem uma distribuição em toda a faixa de tamanhos. A seguir, a Figura 13 ilustra essa distribuição. Enquanto 80% dos criptogramas de Curvas Elípticas situam-se na faixa de até 100 mil *bits*, para o algoritmo ElGamal situam-se na faixa de até 350 mil *bits*.

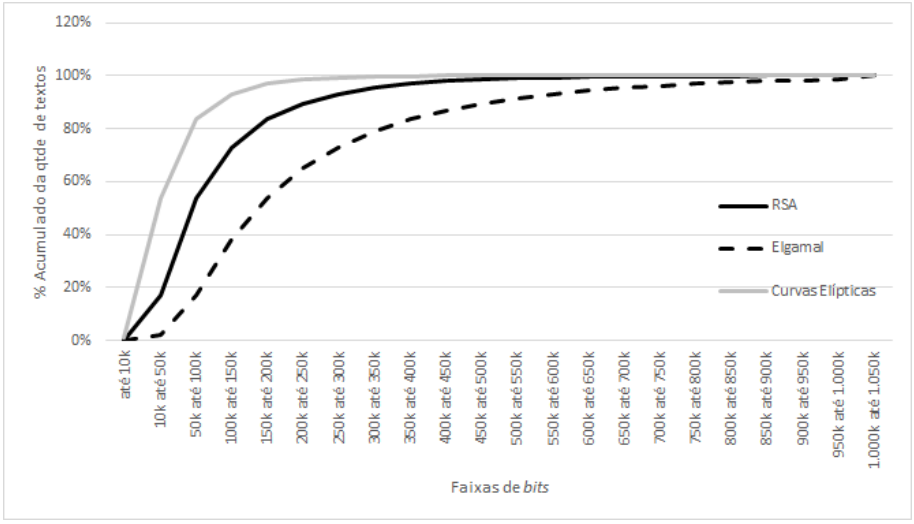


Figura 13 – Curva de percentual acumulado - quantidade de criptogramas por tamanho de *bits*

4.1.3 Etapa 3 - Vetorização

Na fase de pré-processamento, os criptogramas foram transformados em vetores. Como descrito no Capítulo 3 e ilustrado na Figura 13, os criptogramas gerados pelos diferentes algoritmos assimétricos possuem tamanhos diferentes. Para poder comparar todos os vetores de forma imparcial, sem um possível viés que ajudaria na classificação dos criptogramas, foi necessário selecionar uma mesma quantidade de *bits*, nos criptogramas dos três algoritmos, para formar o vetor correspondente.

Como também citado na seção1.5, um dos resultados deste trabalho é identificar como se comporta a taxa de acerto dos classificadores conforme a quantidade de *bits* dos criptogramas diminui, para determinar a quantidade mínima necessária para uma identificação do algoritmo criptográfico.

Devido ao perfil de tamanho dos criptogramas das Curvas Elípticas, evidenciado na Figura 13, iremos selecionar 2.000 amostras de cada base de criptogramas com tamanhos variando entre 4076, 10.000, 50.000, 100.000 e 150.000 *bits*. O valor de 4076 *bits* refere-se ao menor criptograma obtido pelas cifras utilizadas nesta pesquisa. Não foi necessário reduzir mais o tamanho dos criptogramas, pois, como está mostrado na seção 4.4, os classificadores atingiram resultados igual a um acerto aleatório para o tamanho de 4076

bits. Esta mesma quantidade de criptogramas (2.000 amostras) foi utilizada também por Manjula e Anitha(16). Uma visão esquemática do processo descrito nessa seção está representada na Figura 14.

É importante destacar que, para manter a premissa de minimizar possível viés nos dados, a seleção da *string* de *bits* foi diferente para cada criptograma. Durante o processo de vetorização, o início da sequência de *bits* foi atribuído à uma variável aleatória. Portanto, para cada criptograma, a *string* de *bits* inicia-se em uma posição diferente do criptograma.

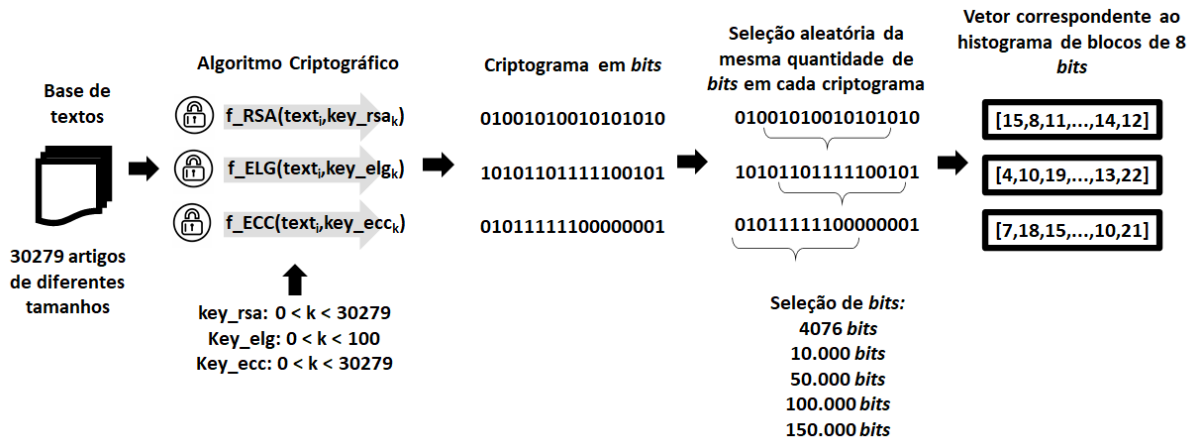


Figura 14 – Processo para seleção dos bits que serão vetorizados

4.1.4 Etapa 4 - Construção das bases de vetores

Para a realização dos experimentos, são construídas 2 bases de vetores para cada algoritmo criptográfico:

1. Base de treinamento e teste: As 2.000 amostras de vetores de cada um dos três algoritmos, em cada faixa de tamanho, foram misturadas e separadas em dois subgrupos na proporção $\frac{2}{3}$ para treinamento e $\frac{1}{3}$ para testes. Segundo Stapor et al.(42), quanto maior o *dataset*, menor o viés, porém, maior a variância. Os autores concluem que o ponto de equilíbrio adequado para *trade-off* entre variância de viés pode ser obtido pela proporção de 66% e 33% entre treinamento e teste.

2. Base de Validação: A base de validação é uma segunda base composta por outras 2.000 amostras de vetores dos três criptogramas geradas por outras chaves aleatórias.

4.1.5 Etapa 5 - Processo de classificação

Conforme indicado no capítulo 2, foram escolhidos os algoritmos de KNN, *Random Forest* e SVM. A procura por otimizações nos hiper-parâmetros dos algoritmos não estava

no escopo desta pesquisa (destacado como sugestão de trabalhos futuros), portanto, os classificadores foram implementados considerando as configurações padrão utilizadas pela biblioteca *scikit learn* e referenciadas nos trabalhos descritos na tabela 7.

Algoritmos	hiper-parâmetros	Referência
SVM	C = 1 Kernel - *Radial Basis Function* $\gamma = \frac{1}{(256 \times var(vetor))}$ Tolerância = 0,001	Chang e Lin(43)
Random Forest	Número de estimadores = 100 Critérios para divisão: Função gini Mínimo de separação: 2 (árvore binária) Mínimo de folhas: 1	Breiman(44)
KNN	Número de vizinhos = 5 peso uniforme	Bentley(45) Omohundro(46)

Tabela 7 – Hiper-parâmetros dos classificadores

Após o treinamento dos modelos utilizando a respectiva base, o classificador é submetido à tarefa de predizer a classe da base de testes.

4.1.6 Etapa 6 - Resultados e Validação

Nesta etapa do experimento, os resultados da predição da base de testes (medidas de desempenho descritas no Capítulo 3) são calculados e validados por uma segunda rodada de classificação, utilizando a base de validação. Espera-se que os resultados sejam os mesmos da etapa de testes, validando o desempenho dos classificadores.

4.2 Cenários dos experimentos

Três cenários utilizando os criptogramas foram estudados de forma a identificar características que nos ajudem a reconhecer os padrões e o comportamento de cada algoritmo criptográfico, conforme Figura 15:

Cenário 1 Os criptogramas foram comparados aos pares. Nesta etapa foi possível avaliar se os padrões de cada algoritmo serão suficientes para uma distinção entre criptogramas.

Cenário 2 Os criptogramas dos três algoritmos foram utilizados simultaneamente. Desta forma foi possível avaliar se os resultados dos classificadores continuarão os mesmos em um ambiente único.

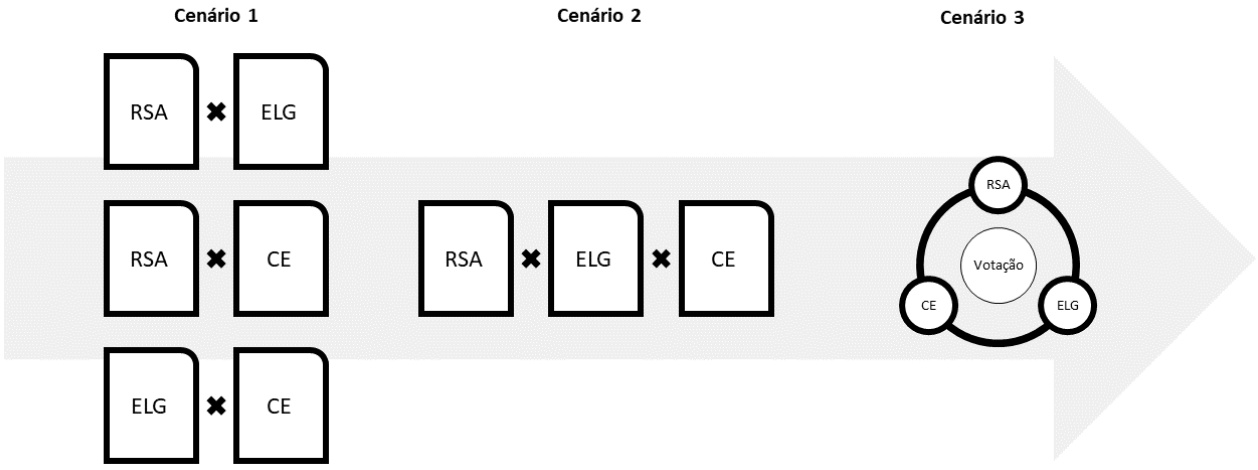


Figura 15 – Cenários comparativos de análise dos criptogramas

Cenário 3 O cenário de votação com os algoritmos de aprendizado de máquina. Espera-se com esta etapa verificar se o desempenho dos classificadores é otimizado utilizando o esquema de votação pela maioria.

Em cada cenário, todos os experimentos foram repetidos para cada faixa de tamanho de *bits*.

4.3 Considerações sobre a repetição dos experimentos

Segundo a teoria do planejamento estatístico de experimentos (47), o cálculo da quantidade de repetições é realizado pela fórmula:

$$r \geq 2(Z_\alpha + Z_\beta)^2(\frac{\delta}{\sigma})^2 \tag{4.1}$$

Onde:

- A potência $(1 - \beta)$: a probabilidade de declarar a existência do efeito se esse efeito existir;
- A significância (α): a incerteza da inferência gerada pelo processo;
- O erro experimental (δ): a grandeza da diferença que deve ser detectada;

- O desvio padrão(σ): Calculado com base em uma estimativa da variância do erro experimental (σ^2).

O desvio padrão que foi adotado aqui será a maior média do desvio padrão das acurácias, nos cenários de comparação dois a dois e dos três algoritmos juntos. Inicialmente foram realizadas 10 repetições de cada experimento (compreendendo as etapas 1 a 6 destacadas na Figura 9). O maior valor de desvio padrão, foi alcançado com os dados de teste, no cenário 1 de comparações (dois a dois).

Considerando uma potência de teste de 95%, uma significância de 5%, o erro experimental como 1%, e o desvio padrão igual a 0,011, o número de repetições necessárias é 32. Portanto, os resultados apresentados a seguir são a média simples de 32 repetições do mesmo experimento, considerando cada tamanho de criptograma e cenário de comparação.

4.4 Resultados

Nesta primeira etapa, os criptogramas foram comparados dois a dois, com o objetivo de verificar se os classificadores os distinguem. Os resultados a seguir das matrizes de confusão, provenientes do teste dos classificadores com $\frac{1}{3}$ da base de amostras (base de testes), foram agrupados de acordo com a quantidade de *bits* e são a média simples de 32 repetições, conforme indicado em 4.3.

Além disso, as análises e resultados em cada cenário de comparação são descritos em ordem decrescente do tamanho do criptograma, partindo dos melhores resultados encontrados. Desta forma, será possível identificar a faixa de *bits* em que não será possível distinguir os criptogramas.

4.4.1 Cenário 1 (a) - RSA x Curvas Elípticas

Neste primeiro cenário comparativo, os criptogramas do RSA e Curvas Elípticas são analisados. Na comparação de vetores com 150.000 *bits* (Tabela 9), o classificador SVM foi o de melhor desempenho, conseguindo distinguir 80% dos criptogramas RSA e das Curvas Elípticas. Nas demais métricas utilizadas - Precisão, *Recall* e F1, o SVM também foi superior aos demais classificadores.

Guardada as devidas proporções, pois não foi utilizada a mesma base de textos em claro e chaves, os resultados estão na mesma ordem de grandeza dos resultados alcançados por Manjula e Anitha(16), que conseguiu distinguir os criptogramas do RSA e Curvas Elípticas com uma acurácia de 73,2% e 74,3% respectivamente, utilizando árvores de decisão. Destaca-se que também foram avaliadas cifras simétricas. Outro resultado que podemos citar deste trabalho é que para criptogramas entre 10.000 e 100.000 *bytes*, que

seria entre 80.000 e 800.000 *bits*, a taxa de acurácia de acertos de todos os criptogramas (simétricos e assimétricos) foi da ordem de 76,3%.

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	CE	RSA	CE	RSA	CE
RSA	523	131	443	219	328	335
CE	138	527	180	479	183	475

Tabela 8 – Matriz de Confusão - RSA x CE - 150.000 *bits*

150.000 *bits*		RSA			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	80%	79%	80%	80%	80%	79%	80%
Random Forest	70%	71%	67%	69%	69%	73%	71%
KNN	61%	64%	50%	56%	59%	72%	65%

Tabela 9 – Cálculo dos Resultados - RSA x CE - 150.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	CE	RSA	CE	RSA	CE
RSA	492	168	410	250	390	273
CE	164	496	211	449	262	394

Tabela 10 – Matriz de Confusão - RSA x CE - 100.000 *bits*

100.000 *bits*		RSA			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	75%	75%	75%	75%	75%	75%	75%
Random Forest	65%	66%	62%	64%	64%	68%	66%
KNN	59%	60%	59%	59%	59%	60%	60%

Tabela 11 – Cálculo dos Resultados - RSA x CE - 100.000 *bits*

À medida em que a quantidade de *bits* diminui, verifica-se a redução da acurácia dos classificadores, como esperado, atingindo entre 50% e 53% no cenário de 4076 *bits* (próximo a um acerto aleatório). Este comportamento sugere a comprovação da hipótese dessa pesquisa: a existência de padrões nas cifras assimétricas, possibilitando o treinamento de algoritmos de aprendizado. Estes padrões são mais evidentes proporcionalmente ao tamanho dos criptogramas.

É possível identificar a faixa dos 50 mil *bits* como a faixa mínima para um desempenho aceitável de classificação (66%). Após essa faixa, reduzindo para 10 mil *bits*, o desempenho cai para 56%, estando mais próximo a um acerto aleatório.

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	CE	RSA	CE	RSA	CE
RSA	436	226	358	302	396	267
CE	219	439	260	400	349	308

Tabela 12 – Matriz de Confusão - RSA x CE - 50.000 *bits*

50.000 *bits*		RSA			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	66%	67%	66%	66%	66%	67%	66%
Random Forest	57%	58%	54%	56%	57%	61%	59%
KNN	53%	53%	60%	56%	54%	47%	50%

Tabela 13 – Cálculo dos Resultados - RSA x CE - 50.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	CE	RSA	CE	RSA	CE
RSA	361	301	324	336	354	305
CE	286	372	295	365	345	316

Tabela 14 – Matriz de Confusão - RSA x CE - 10.000 *bits*

10.000 *bits*		RSA			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	56%	56%	55%	55%	55%	57%	56%
Random Forest	52%	52%	49%	51%	52%	55%	54%
KNN	51%	51%	54%	52%	51%	48%	49%

Tabela 15 – Cálculo dos Resultados - RSA x CE - 10.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	CE	RSA	CE	RSA	CE
RSA	348	309	304	357	311	350
CE	315	348	296	363	314	345

Tabela 16 – Matriz de Confusão - RSA x CE - 4076 *bits*

4076 *bits*		RSA			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	53%	52%	53%	53%	53%	52%	53%
Random Forest	51%	51%	46%	48%	50%	55%	53%
KNN	50%	50%	47%	48%	50%	52%	51%

Tabela 17 – Cálculo dos Resultados - RSA x CE - 4076 *bits*

4.4.2 Cenário 1 (b) - ElGamal x Curvas Elípticas

Os resultados da comparação entre ElGamal e Curvas Elípticas seguem o mesmo perfil identificado no cenário RSA e Curvas Elípticas. O algoritmo de aprendizado SVM continua com resultados melhores dentre os classificadores e os resultados variam entre 52% e 80%, semelhante ao cenário com o RSA. O limiar de 50 mil *bits* também é identificado nesse cenário de comparação.

	Predição SVM		Predição RF		Predição KNN	
Real	ELG	CE	ELG	CE	ELG	CE
ELG	526	134	447	213	392	272
CE	127	532	177	483	238	418

Tabela 18 – Matriz de Confusão - ELG x CE - 150.000 *bits*

150.000 *bits*		ELG			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	80%	81%	80%	80%	80%	81%	80%
Random Forest	70%	72%	68%	70%	69%	73%	71%
KNN	61%	62%	59%	61%	61%	64%	62%

Tabela 19 – Cálculo dos Resultados - ELG x CE - 150.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	ELG	CE	ELG	CE	ELG	CE
ELG	491	167	406	253	389	272
CE	164	499	211	450	293	366

Tabela 20 – Matriz de Confusão - ELG x CE - 100.000 *bits*

100.000 *bits*		ELG			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	75%	75%	75%	75%	75%	75%	75%
Random Forest	65%	66%	62%	64%	64%	68%	66%
KNN	57%	57%	59%	58%	57%	56%	56%

Tabela 21 – Cálculo dos Resultados - ELG x CE - 100.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	ELG	CE	ELG	CE	ELG	CE
ELG	447	212	365	294	374	285
CE	222	439	256	405	316	346

Tabela 22 – Matriz de Confusão - ELG x CE - 50.000 *bits*

50.000 *bits*		ELG			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	67%	68%	67%	67%	66%	65%	67%
Random Forest	58%	59%	55%	57%	58%	61%	60%
KNN	55%	54%	57%	55%	55%	52%	54%

Tabela 23 – Cálculo dos Resultados - ELG x CE - 50.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	ELG	CE	ELG	CE	ELG	CE
ELG	362	297	326	334	371	290
CE	283	378	293	367	364	294

Tabela 24 – Matriz de Confusão - ELG x CE - 10.000 *bits*

10.000 *bits*		ELG			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	56%	56%	55%	56%	56%	57%	57%
Random Forest	52%	53%	49%	51%	52%	56%	54%
KNN	50%	50%	56%	53%	50%	45%	47%

Tabela 25 – Cálculo dos Resultados - ELG x CE - 10.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	ELG	CE	ELG	CE	ELG	CE
ELG	343	314	303	358	296	365
CE	323	340	298	361	296	363

Tabela 26 – Matriz de Confusão - ELG x CE - 4076 *bits*

4076 *bits*		ELG			CE		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	52%	52%	52%	52%	52%	51%	52%
Random Forest	50%	50%	46%	48%	50%	55%	52%
KNN	50%	50%	45%	47%	50%	55%	52%

Tabela 27 – Cálculo dos Resultados - ELG x CE - 4076 *bits*

4.4.3 Cenário 1 (c) - RSA x ElGamal

Neste cenário, pela análise dos valores de acurácia em todos os cenários de variação de *bits*, verificou-se que não foi possível obter um classificador que possa distinguir criptogramas RSA e ElGamal com uma taxa de acerto superior a um acerto aleatório.

A própria característica dos criptogramas pode ser uma fonte de explicação para a maior identificação das Curvas Elípticas. Enquanto os criptogramas do RSA e ElGamal são semelhantes (exponenciações modulares), os criptogramas das curvas elípticas são pontos cartesianos. Esta diferença pode ter sido identificada nos algoritmos de classificação, possibilitando a acurácia maior para o caso das Curvas Elípticas.

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	ELG	RSA	ELG	RSA	ELG
RSA	327	332	309	349	264	400
ELG	325	336	313	348	266	390

Tabela 28 – Matriz de Confusão - RSA x ELG - 150.000 *bits*

150.000 *bits*		RSA			ELG		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	50%	50%	50%	50%	50%	51%	51%
Random Forest	50%	50%	47%	48%	50%	53%	51%
KNN	50%	50%	40%	44%	49%	59%	54%

Tabela 29 – Cálculo dos Resultados - RSA x ELG - 150.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	ELG	RSA	ELG	RSA	ELG
RSA	335	324	301	363	327	329
ELG	321	340	294	362	330	335

Tabela 30 – Matriz de Confusão - RSA x ELG - 100.000 *bits*

100.000 *bits*		RSA			ELG		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	51%	51%	51%	51%	51%	51%	51%
Random Forest	50%	51%	45%	48%	50%	55%	52%
KNN	50%	50%	50%	50%	50%	50%	50%

Tabela 31 – Cálculo dos Resultados - RSA x ELG - 100.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	ELG	RSA	ELG	RSA	ELG
RSA	333	327	306	354	359	299
ELG	334	327	301	359	365	297

Tabela 32 – Matriz de Confusão - RSA x ELG - 50.000 *bits*

50.000 *bits*		RSA			ELG		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	50%	50%	50%	50%	50%	49%	50%
Random Forest	50%	50%	46%	48%	50%	54%	52%
KNN	50%	50%	55%	52%	50%	45%	47%

Tabela 33 – Cálculo dos Resultados - RSA x ELG - 50.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	ELG	RSA	ELG	RSA	ELG
RSA	326	335	298	368	307	355
ELG	334	325	286	368	306	352

Tabela 34 – Matriz de Confusão - RSA x ELG - 10.000 *bits*

10.000 *bits*		RSA			ELG		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	49%	49%	49%	49%	49%	49%	49%
Random Forest	50%	51%	45%	48%	50%	56%	53%
KNN	50%	50%	46%	48%	50%	53%	52%

Tabela 35 – Cálculo dos Resultados - RSA x ELG - 10.000 *bits*

	Predição SVM		Predição RF		Predição KNN	
Real	RSA	ELG	RSA	ELG	RSA	ELG
RSA	325	336	301	358	335	324
ELG	335	325	307	353	341	321

Tabela 36 – Matriz de Confusão - RSA x ELG - 4076 *bits*

4076 *bits*		RSA			ELG		
Classificador	Acurácia	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	49%	49%	49%	49%	49%	49%	49%
Random Forest	50%	50%	46%	48%	50%	53%	51%
KNN	50%	50%	51%	50%	50%	48%	49%

Tabela 37 – Cálculo dos Resultados - RSA x ELG - 4076 *bits*

4.4.4 Cenário 2 - RSA x ElGamal x CE

Neste segundo cenário de experimentos, todos os criptogramas foram estudados de uma só vez. Foi possível avaliar a variação do desempenho dos classificadores frente ao cenário anterior de comparações dois a dois e também avaliar se a estratégia de construir um classificador único é melhor do que um classificador binário.

Os resultados representados nas tabelas a seguir mostram que ainda é possível treinar e testar os algoritmos de classificação com os três criptogramas. Todos os resultados de acurácia foram maiores que uma taxa aleatória de acerto (que nesse cenário é de 33%).

Os mesmos resultados alcançados no Cenário 1 também puderam ser identificados nesse cenário: A maior identificação dos criptogramas de Curvas Elípticas e o desempenho superior do classificador SVM. Isso mostra a consistência dos resultados alcançados.

	Predição SVM			Predição RF			Predição KNN		
Real	RSA	ELG	CE	RSA	ELG	CE	RSA	ELG	CE
RSA	267	288	113	231	264	170	119	272	271
ELG	264	286	106	236	260	162	125	269	261
CE	83	74	500	121	129	408	78	178	406

Tabela 38 – Matriz de Confusão - RSA x ELG x CE - 150.000 *bits*

150.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	53%	43%	40%	42%	44%	44%	44%	70%	76%	73%
RF	45%	39%	35%	37%	40%	39%	40%	55%	62%	58%
KNN	40%	37%	18%	24%	37%	41%	39%	43%	61%	51%

Tabela 39 – Cálculo dos Resultados - RSA x ELG x CE - 150.000 *Bits*

	Predição SVM			Predição RF			Predição KNN		
Real	RSA	ELG	CE	RSA	ELG	CE	RSA	ELG	CE
RSA	270	262	135	234	245	182	160	246	251
ELG	255	267	138	227	251	184	153	253	258
CE	98	97	459	143	152	361	105	203	352

Tabela 40 – Matriz de Confusão - RSA x ELG x CE - 100.000 *bits*

100.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	50%	43%	40%	42%	43%	41%	42%	63%	70%	66%
RF	43%	39%	35%	37%	39%	38%	38%	50%	55%	52%
KNN	39%	38%	24%	30%	36%	38%	37%	41%	53%	46%

Tabela 41 – Cálculo dos Resultados - RSA x ELG x CE - 100.000 *Bits*

	Predição SVM			Predição RF			Predição KNN		
Real	RSA	ELG	CE	RSA	ELG	CE	RSA	ELG	CE
RSA	243	242	173	218	237	207	171	230	256
ELG	245	248	168	216	235	207	174	238	249
CE	132	135	394	179	189	292	158	207	297

Tabela 42 – Matriz de Confusão - RSA x ELG x CE - 50.000 *bits*

50.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	45%	39%	37%	38%	40%	38%	39%	54%	60%	56%
RF	38%	36%	33%	34%	36%	36%	36%	41%	44%	43%
KNN	36%	34%	26%	29%	35%	36%	36%	37%	45%	41%

Tabela 43 – Cálculo dos Resultados - RSA x ELG x CE - 50.000 *Bits*

	Predição SVM			Predição RF			Predição KNN		
Real	RSA	ELG	CE	RSA	ELG	CE	RSA	ELG	CE
RSA	222	226	213	209	228	221	140	258	263
ELG	227	219	211	206	228	228	143	249	269
CE	191	178	293	200	214	246	139	248	271

Tabela 44 – Matriz de Confusão - RSA x ELG x CE - 10.000 *bits*

10.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	37%	35%	34%	34%	35%	33%	34%	41%	44%	42%
RF	35%	34%	32%	33%	34%	34%	34%	35%	37%	36%
KNN	33%	33%	21%	26%	33%	38%	35%	34%	41%	37%

Tabela 45 – Cálculo dos Resultados - RSA x ELG x CE - 10.000 *Bits*

	Predição SVM			Predição RF			Predição KNN		
Real	RSA	ELG	CE	RSA	ELG	CE	RSA	ELG	CE
RSA	219	232	218	202	221	239	135	202	326
ELG	220	219	214	202	216	240	136	201	326
CE	197	215	246	200	219	241	130	203	321

Tabela 46 – Matriz de Confusão - RSA x ELG x CE - 4076 *bits*

4076 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
SVM	35%	34%	33%	34%	33%	34%	33%	36%	37%	37%
RF	33%	33%	31%	32%	33%	33%	33%	33%	36%	35%
KNN	33%	34%	20%	25%	33%	30%	32%	33%	49%	39%

Tabela 47 – Cálculo dos Resultados - RSA x ELG x CE - 4076 *Bits*

4.4.5 Cenário 3 - Votação - RSA x ElGamal x CE

O terceiro cenário desta pesquisa busca verificar se a combinação das classificações dos três algoritmos de aprendizado produz taxas melhores de classificação. Espera-se que as classificações feitas incorretamente nos cenários anteriores possam ser corrigidas ao se estabelecer uma votação entre os três classificadores.

Como exemplo, nesse experimento, um mesmo vetor da base de testes é classificado pelos três algoritmos de aprendizado e se no final, pelo menos 2 deles definem que este possui a classificação RSA, então a resposta desse classificador combinado será RSA.

Os resultados deste cenário mostraram que os classificadores combinados não conseguem obter um resultado melhor do que o cenário 2. Se analisarmos pela métrica da acurácia, o melhor resultado de 48% para 150 mil *bits* ainda é pior que a acurácia no cenário 2, que é de 53%.

Também é possível identificar que resultados do F1*Score* para o RSA foram piores em todos os tamanhos de criptogramas avaliados, chegando a ser menores que o valor aleatório de 33%. Portanto, a premissa de que o erro em um classificador poderia ser corrigido pelos dois outros classificadores não está correta para as condições adotadas nessa pesquisa.

Os resultados alcançados são explicados pelo baixo número de classificadores utilizados no comitê (3 classificadores) e pelo esquema de composição votação (votação pela maioria). Entende-se que, para o cenário do comitê conseguir resultados melhores que os resultados do cenário comparativo anterior, seria necessário utilizar também outros classificadores (Naïve Bayes, Redes Neurais e Árvores de Decisão por exemplo), aumentando a quantidade de classificadores utilizados e investigar quais os melhores esquemas de composição de votos para este caso.

	Predição Comitê		
Real	RSA	ELG	CE
RSA	166	252	241
ELG	179	250	230
CE	52	78	532

Tabela 48 – Matriz de Confusão - Comitê - RSA x ELG x CE - 150.000 *bits*

150.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
Comitê	48%	72%	25%	37%	43%	38%	40%	53%	80%	64%

Tabela 49 – Cálculo dos Resultados - Comitê - RSA x ELG x CE - 150.000 *Bits*

	Predição Comitê		
Real	RSA	ELG	CE
RSA	189	224	248
ELG	184	228	247
CE	70	97	494

Tabela 50 – Matriz de Confusão - Comitê - RSA x ELG x CE - 100.000 *bits*

100.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
Comitê	46%	75%	29%	41%	42%	35%	38%	50%	75%	60%

Tabela 51 – Cálculo dos Resultados- Comitê - RSA x ELG x CE - 100.000 *Bits*

	Predição Comitê		
Real	RSA	ELG	CE
RSA	167	212	282
ELG	170	214	271
CE	106	133	425

Tabela 52 – Matriz de Confusão - Comitê - RSA x ELG x CE - 50.000 *bits*

50.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
Comitê	41%	60%	25%	36%	38%	33%	35%	43%	64%	52%

Tabela 53 – Cálculo dos Resultados - Comitê - RSA x ELG x CE - 50.000 *Bits*

	Predição Comitê		
Real	RSA	ELG	CE
RSA	143	197	320
ELG	148	193	321
CE	129	167	363

Tabela 54 – Matriz de Confusão - Comitê - RSA x ELG x CE - 10.000 *bits*

10.000 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
Comitê	35%	52%	22%	31%	35%	29%	32%	36%	55%	44%

Tabela 55 – Cálculo dos Resultados - Comitê - RSA x ELG x CE - 10.000 *Bits*

	Predição Comitê		
Real	RSA	ELG	CE
RSA	144	176	341
ELG	147	170	345
CE	130	168	359

Tabela 56 – Matriz de Confusão - Comitê - RSA x ELG x CE - 4076 *bits*

4076 *bits*		RSA			ELG			CE		
Class.	ACC	Precisão	*Recall*	F1	Precisão	*Recall*	F1	Precisão	*Recall*	F1
Comitê	34%	52%	22%	31%	33%	26%	29%	34%	55%	42%

Tabela 57 – Cálculo dos Resultados - Comitê - RSA x ELG x CE - 4076 *Bits*

4.5 Validação dos resultados e avaliação do *Overtraining* dos modelos

Seguindo o experimento apresentado na Figura 9, os classificadores foram executados em seguida à etapa de testes, considerando uma segunda base disjunta de criptogramas com chaves diferentes da base de treinamento e testes. Adicionalmente, durante a etapa de testes, foram capturados também os resultados dos modelos treinados predizendo para as amostras utilizadas no treinamento. O objetivo é verificar o *overtraining* dos classificadores e seu impacto na capacidade de generalização dos modelos.

Os gráficos a seguir ilustram a variação da acurácia nos diversos cenários de comparação, considerando os dados de Treinamento, Teste e Validação.

Foi possível identificar que o classificador SVM, para poder determinar as fronteiras de decisão, erra até 8% das amostras indicadas a ele para treinamento (Cenário 2). Também foi possível verificar que, à medida em que os vetores possuem mais *bits* e o padrão fica mais evidente, as fronteiras ficam mais bem definidas, necessitando de uma quantidade menor de amostras dentro da margem entre os hiperplanos de classificação.

Para o classificador *Random Forest*, como relatado no capítulo 3, seu método de predição utiliza diversas árvores de decisão. Portanto não é necessário renunciar a alguma amostra do conjunto de treinamento para poder definir seu modelo de predição. Essa informação pode ser identificada nos gráficos, onde a acurácia do modelo para os dados de treinamento é sempre 100%.

Em relação ao classificador KNN, a abordagem de classificação pela proximidade mostrou-se inferior em relação aos demais métodos de aprendizado. A justificativa para este resultado é encontrada no trabalho de Kouiroukidis e Evangelidis(48). Segundo os autores, em espaços dimensionais elevados, as distâncias entre os pontos mais próximos e mais distantes dos exemplos que se quer predizer a classificação tornam-se quase iguais.

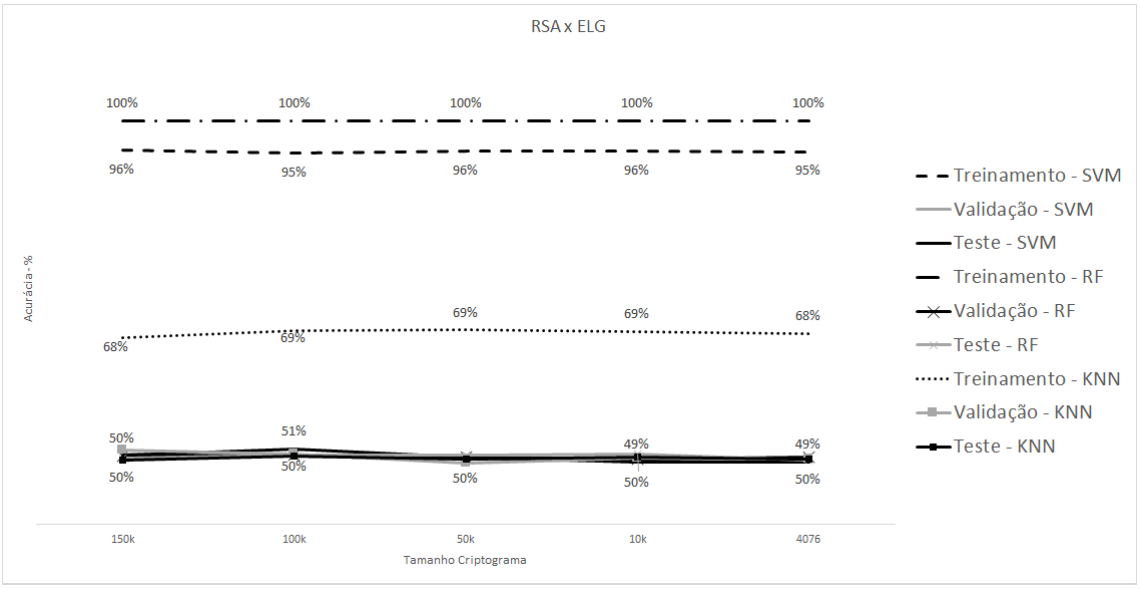


Figura 16 – Comparação RSAxELG - Acurácia

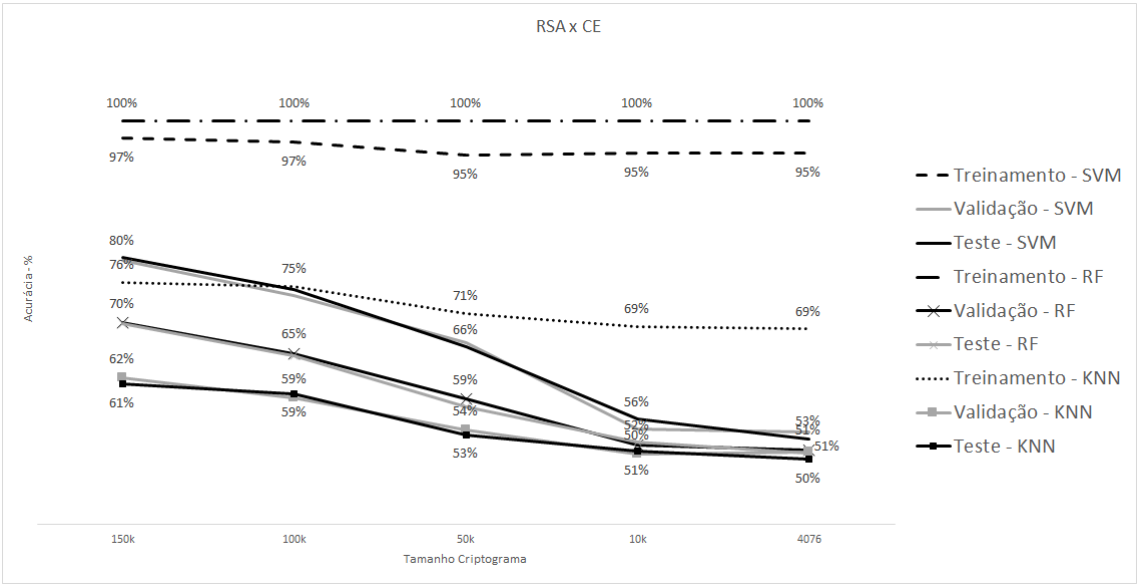


Figura 17 – Comparação RSAxCE - Acurácia

Nesta pesquisa, o espaço dimensional é de 256 dimensões (2^8).

Em termos da acurácia na base de validação, como é possível verificar nos gráficos, os resultados são iguais aos resultados de teste, comprovando os resultados obtidos pelos experimentos.

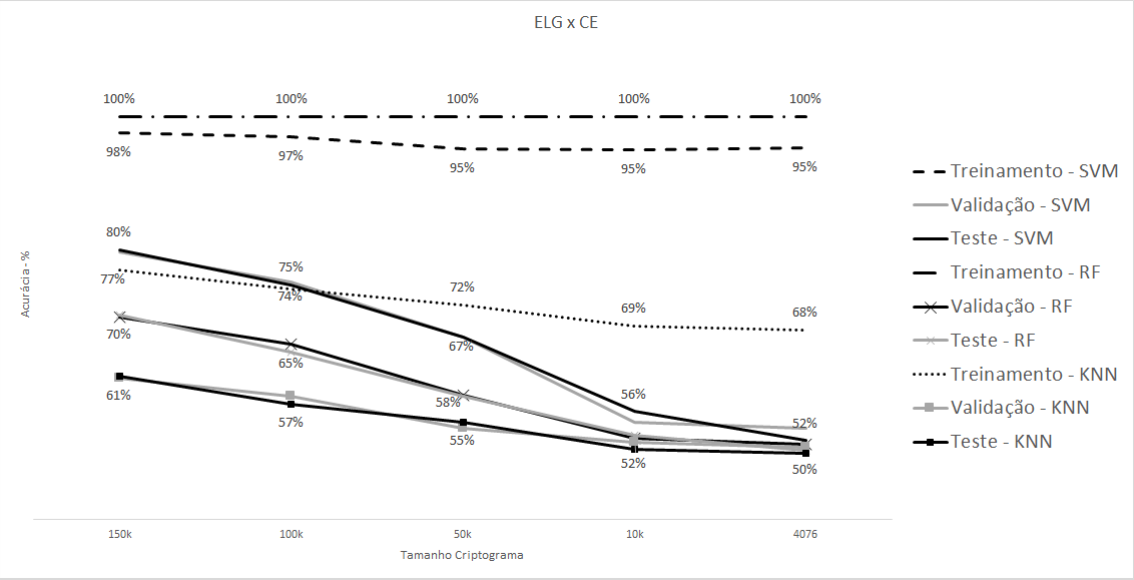


Figura 18 – Comparação ELGxCE - Acurácia

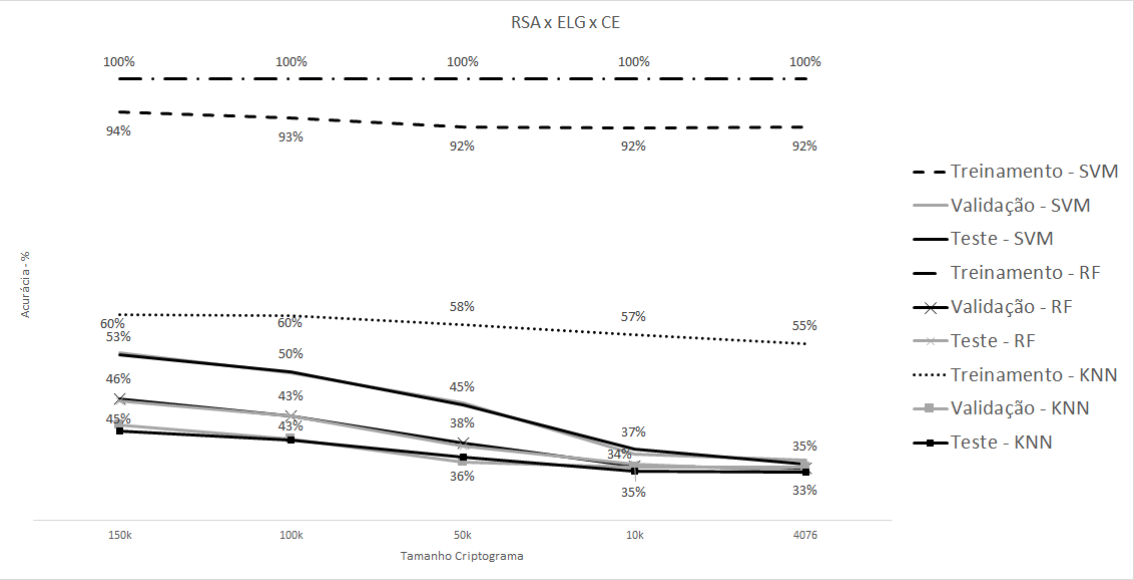


Figura 19 – Comparação RSAxELGxCE - Acurácia

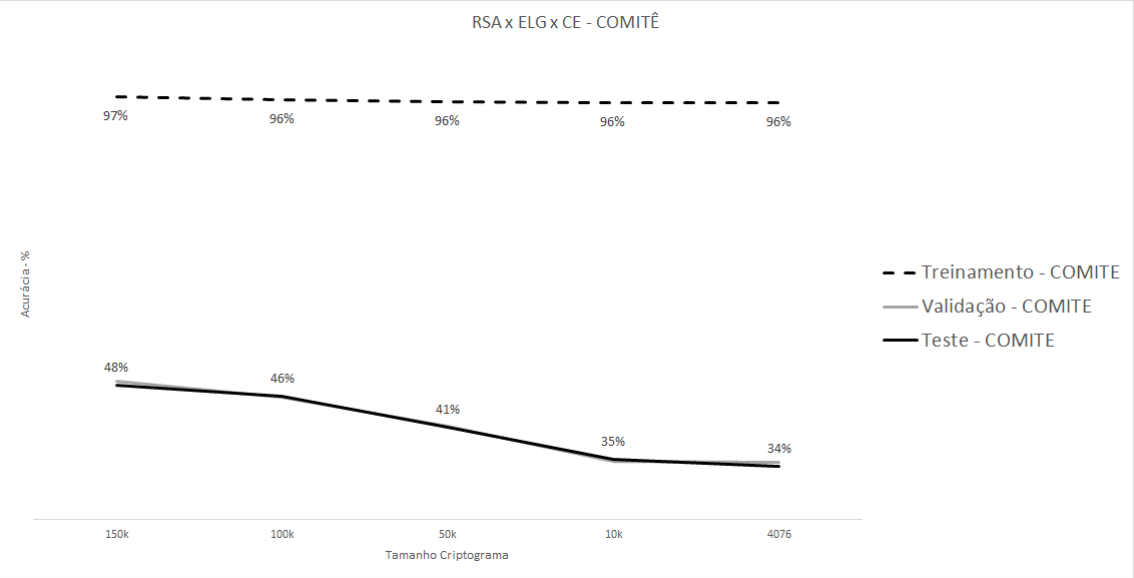


Figura 20 – Comparação Comitê - RSAxELGxCE - Acurácia

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

De acordo com os resultados apresentados no capítulo 4 e resgatando a hipótese e objetivo desta pesquisa:

Hipótese: Existem padrões nos criptogramas gerados pelo algoritmo ElGamal que podem ser identificados pelos algoritmos de aprendizado de máquina.

Objetivo: O objetivo desta pesquisa foi avaliar a ocorrência de padrões em criptogramas gerados pelos algoritmos assimétricos RSA, ElGamal e Curvas Elípticas de forma que seja possível classificar um novo criptograma utilizando os algoritmos de aprendizado SVM, *Random Forest* e *K-Nearest Neighbor*.

Foi possível verificar a comprovação desta hipótese e o alcance do objetivo geral desta pesquisa. Foram identificados padrões nos criptogramas gerados pelo RSA e Curvas Elípticas, identificados também pelos autores anteriores, e nos criptogramas gerados pela cifra ElGamal, de forma que fosse possível treinar os algoritmos de aprendizado na tarefa de classificação de novos criptogramas. Identificou-se também nesta pesquisa o alto grau de semelhança entre os criptogramas RSA e ElGamal, impossibilitando uma distinção entre os dois e um padrão característico nos criptogramas gerados pelas curvas elípticas, possibilitando sua identificação.

O algoritmos de aprendizado de máquina *Support Vector Machine* alcançou os melhores resultados em todos os cenários dos experimentos realizados nessa pesquisa, reforçando sua ampla utilização pelos pesquisados da área, conforme apontado na Figura 2. O classificador *Random Forest* obteve resultados menores que o SVM, porém maiores que o *K-Nearest Neighbor*.

Como contribuições desta pesquisa, além da identificação de padrões nos criptogramas ElGamal, pode-se destacar o estudo exclusivo com cifras assimétricas em um ambiente de múltiplas chaves e a análise do desempenho dos classificados à medida que se reduz o tamanho dos criptogramas, em que foi observado o tamanho de 50 mil *bits* como o tamanho mínimo para uma taxa de acerto não ser caracterizada como aleatória.

5.2 Trabalhos Futuros

Para este autor, o campo de estudo da identificação de padrões em criptogramas ainda apresenta diversas oportunidades de estudo. Pode-se citar:

1. Um dos problemas identificados pela pesquisa bibliográfica é a necessidade da criação de uma base única de criptogramas para que os resultados de pesquisa possam ser comparados. Como foi citado no capítulo 2, as pesquisas adotaram diversas bases de textos em claro diferentes, muita das vezes sem menção nos artigos, para realizar os experimentos de identificação. Esta base universal deve conter criptogramas dos mais diversos algoritmos criptográficos (simétricos e assimétricos), tamanhos de chaves e modos de operação (casos de cifras simétricas). Deve estar disponível e em diversos formatos de banco de dados.

2. Pela utilização crescente das Curvas Elípticas, uma oportunidade de estudo seria explorar se há padrões característicos específicos nas diversas curvas elípticas utilizadas: NIST, Brainpool, Hessian, Edwards, etc.

3. Outra variável interessante a ser explorada em trabalhos futuros é verificar se há variações nos padrões de acordo com as linguagens de programação e bibliotecas utilizadas. Por exemplo, uma biblioteca em python e em C que criptografam o mesmo algoritmo, com a mesma chave, terá o mesmo criptograma, ou terá diferenças que possam ser identificadas?

4. A redução de dimensionalidade é outro tema importante nessa área de pesquisa. Os criptogramas são transformados em vetores de 256 posições (para dicionários de blocos de 8 *bits*). Durante esta pesquisa, foi verificada uma alta correlação entre as 256 posições, indicando uma possível redução de dimensionalidade. Frente a este cenário, qual será o impacto no desempenho dos classificadores, caso seja reduzida a dimensão? Quantas dimensões são necessárias para alcançar o ponto ótimo? Possivelmente, o classificador KNN será beneficiado nessa redução.

5. Dada a pesquisa bibliográfica realizada e à limitação desta pesquisa, considera-se que a otimização dos parâmetros dos classificadores ainda é área de estudo a ser explorada.

6. Assim como foi utilizado pelos autores Manjula e Anitha(16), é necessário pesquisar novas formas de representar os criptogramas além das medidas de similaridade e frequência dos *bytes* dos criptogramas.

REFERÊNCIAS

1 STALLINGS, W. *Criptografia e Segurança de Redes: princípios e práticas*. 6. ed. [S.l.]: Pearson Education do Brasil, 2015. ISBN 9788543014500. 9, 32, 33

2 GAMA, J.; FACELI, K.; LORENA, A.; CARVALHO, A. D. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011. ISBN 9788521618805. Disponível em: <https://books.google.com.br/books?id=4DwelAEACAAJ>. 9, 37, 38, 40, 42, 45

3 HANKERSON, D.; MENEZES, A.; VANSTONE, S. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, 2004. ISBN 978-1-4419-2929-7. Disponível em: <https://www.springer.com/gp/book/9780387952734>. 9, 49, 50

4 STANDARDIZATION, I. O. for. *ISO/IEC 27001:2013 - Tecnologia da informação — Técnicas de segurança — Sistemas de gestão da segurança da informação — Requisitos*. Iso/iec 27001:2013. International Organization for Standardization, 2013. Disponível em: <https://www.abntcatalogo.com.br/norma.aspx?ID=306580>. 16

5 SCHNEIER, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. [S.l.]: John Wiley Sons, Inc, 2015. (20th Anniversary Hardcover). ISBN 9781119096726. 16, 25, 32

6 KERCKHOFF, A. La cryptographie militaire. *Journal des sciences militarires*, p. 5–38, 1883. Disponível em: <https://www.petitcolas.net/kerckhoffs/index.html>. 16

7 BARKER, E. *NIST SP 800-57: Recommendation for key management: Parte 1 - General*. *National Institute for Standards and Technology*, 2020. Disponível em: <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final>. 16

8 TORRES, R. H. *Desenvolvimento E Análise De Funções Criptográficas Para Otimização Dos Padrões De Dispersão Em Criptogramas*. 116 p. Dissertação (Mestrado) — Instituto Militar de Engenharia, Rio de Janeiro, 2011. 16, 26

9 DILEEP, A. D.; SEKHAR, C. C. Identification of block ciphers using support vector machines. In: IEEE. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. Vancouver: IEEE, 2006. p. 2696–2701. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1716462>. 16, 79, 80, 81

10 KNUDSEN, L. R.; MEIER, W. Correlations in rc6 with a reduced number of rounds. In: GOOS, G.; HARTMANIS, J.; LEEUWEN, J. van; SCHNEIER, B. (Ed.). *Fast Software Encryption*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 94–108. ISBN 978-3-540-44706-1. 17

11 GROUP, N. W. *Internet Official Protocol Standards - The Transport Layer Security (TLS) Protocol*. [S.l.], 2008. 17

12 WAZLAWICK, R. *Metodologia de Pesquisa para Ciência da Computação*. Elsevier Brasil, 2017. ISBN 9788535277838. Disponível em: <https://books.google.com.br/books?id=BZioBQAAQBAJ>. 20

13 SOUZA, W. A. R. D. *Identificação De Padrões Em Criptogramas Usando Técnicas De Classificação De Textos*. 252 p. Dissertação (Mestrado) — Instituto Militar de Engenharia, Rio de Janeiro, 2007. 20, 25, 26, 29

14 OLIVEIRA, C.; SOUZA, W. A. R. D.; XEXEO, J. A. M. Método de agrupamento de criptogramas em função das chaves de cifrar. In: *Workshop em Algoritmos e Aplicações de Mineração de Dados*. Campinas: [s.n.], 2008. Disponível em: <http://homepages.dcc.ufmg.br/~meira/DokuWiki/wiki/_media/waamd_5-2.pdf#page=27>. 20, 25, 26, 28, 29, 79

15 TORRES, R. H.; OLIVEIRA, G. A. de; XEXEO, J. A. M.; SOUZA, W. A. R. D.; LIDEN, R. Detecção de padrões em criptogramas como suporte às atividades de criptoanálise. *Cadernos do IME - Série Informática*, v. 29, p. 38–45, 2010. Disponível em: <https://www.e-publicacoes.uerj.br/index.php/cadinf/article/view/6544>. 20, 25, 26, 79

16 MANJULA, R.; ANITHA, R. Identification of encryption algorithm using decision tree. In: *International Conference on Computer Science and Information Technology*. [s.n.], 2011. p. 237–246. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-17881-8_23>. 20, 25, 26, 28, 29, 52, 55, 71, 79, 80, 81

17 MELLO, F. L. de; XEXEO, J. A. M. Cryptographic algorithm identification using machine learning and massive processing. *IEEE Latin America Transactions*, v. 14, p. 4585–4590, 2016. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7795833>. 20, 25, 27, 28, 29, 30, 78, 79, 80, 81

18 BARBOSA, F. M.; VIDAL, A. R. S. F.; MELLO, F. L. de. Machine learning for cryptographic algorithm identification. *ENIGMA — Brazilian Journal of Information Security and Cryptography*, v. 3, p. 3–8, 2016. Disponível em: <https://enigmajournal.unb.br/index.php/enigma/article/view/55>. 20, 25, 27, 28, 29, 79, 80, 81

19 BARBOSA, F. M.; VIDAL, A. R. S. F.; ALMEIDA, H. L. S. de; MELLO, F. L. de. Machine learning applied to the recognition of cryptographic algorithms used for multimedia encryption. *IEEE Latin America Transactions*, v. 15, p. 1301–1305, 2017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7959350>. 20, 25, 27, 28, 29, 78, 79, 80, 81

20 MELLO, F. L. de; XEXEO, J. A. M. Identifying encryption algorithms in ecb and cbc modes using computational intelligence. *Journal of Universal Computer Science*, v. 24, p. 25–42, 2018. Disponível em: <http://jucs.org/jucs_24_1/identifying_encryption_algorithms_in/jucs_24_01_0025_0042_demello.pdf>. 20, 25, 27, 29, 30, 48, 79, 80, 81

21 BARBOSA, J. C. *Criptografia de chave pública baseada em curvas elípticas*. Dissertação (Mestrado) — COPPE/UFRJ, 2003. 20

22 SOUZA, W. A. R. D. *Identificação De Contextos Lingüísticos Em Linguagens Desconhecidas Geradas Por Cifras De Blocos*. 124 p. Tese (Doutorado) — Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa em Engenharia - COPPE/UFRJ, Rio de Janeiro, 2011. 20

23 MAHESHWARI, P. *Classification of Ciphers*. 28 p. Dissertação (Mestrado) — Indian Institute of Technology, Kanpur, 2001. 24

24 CARVALHO, C. A. B. D. *O uso de técnicas de recuperação de informações em criptoanálise*. Dissertação (Mestrado) — Instituto Militar de Engenharia, Rio de Janeiro, 2006. 26

25 OLIVEIRA, G. A. D. *A Aplicação De Algoritmos Genéticos No Reconhecimento De Padrões Criptográficos*. 93 p. Dissertação (Mestrado) — Instituto Militar de Engenharia, Rio de Janeiro, 2011. 26

26 SINGH, S. *The code book*. [S.l.]: Fourth Estate and Doubleday, 1999. 122-123 p. 31

27 DIFFIE, W.; HELLMAN, M. E. Multiuser cryptographic techniques. In: *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*. New York, NY, USA: Association for Computing Machinery, 1976. (AFIPS '76), p. 109–112. ISBN 9781450379175. Disponível em: <https://doi.org/10.1145/1499799.1499815>. 31

28 DIFFIE, W.; HELLMAN, M. E. New directions in cryptography. *IEEE Transactions on Information Theory*, v. 22, n. 6, p. 644–654, 1976. 32, 33

29 JEVONS, W. S. *The Principles of Science: A Treatise on Logic and Scientific Method*. 1. ed. [S.l.]: Macmillam & Co, 1874. 122-123 p. 32

30 RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 21, n. 2, p. 120–126, fev. 1978. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/359340.359342>. 34

31 ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, v. 31, n. 4, p. 469–472, July 1985. ISSN 1557-9654. 35

32 KOBLITZ, N. Elliptic curve cryptosystems. In: *Mathematics of Computation*. [S.l.: s.n.], 1987. v. 48, p. 203–209. 36

33 MILLER, V. S. Use of elliptic curves in cryptography. In: WILLIAMS, H. C. (Ed.). *Advances in Cryptology — CRYPTO '85 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986. p. 417–426. ISBN 978-3-540-39799-1. 36

34 DEMYTKO, N. A new elliptic curve based analogue of rsa. In: HELLESETH, T. (Ed.). *Advances in Cryptology — EUROCRYPT '93*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 40–49. ISBN 978-3-540-48285-7. 36

35 SUTIKNO, S.; SURYA, A.; EFFENDI, R. An implementation of elgamal elliptic curves cryptosystems. In: *IEEE. APCCAS 1998. 1998 IEEE Asia-Pacific Conference on Circuits and Systems. Microelectronics and Integrating Systems. Proceedings (Cat. No.98EX242)*. [S.l.: s.n.], 1998. p. 483–486. 36

36 CHEN, L.; MOODY, D.; REGENSCHEID, A.; RANDALL, K. *Recommendations for Discrete 4 Logarithm-Based Cryptography: Elliptic Curve Domain Parameters*. [S.l.], 2019. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-186-draft.pdf>. 37

37 MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89–114. ISBN 85-204-168. 38

38 BREIMAN, L. Random forests. *Machine Learning*, v. 45, 2001. Disponível em: <https://doi.org/10.1023/A:1010933404324>. 40, 41

39 BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, Springer, p. 121–167, June 1998. 43

40 SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, v. 45, n. 4, p. 427 – 437, 2009. ISSN 0306-4573. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457309000259>. 46

41 SILVERMAN, R. D. Fast generation of random, strong rsa primes. *RSA Laboratories*, 05 1997. 49

42 STAPOR, K.; KSIENIEWICZ, P.; GARCíA, S.; WOźNIAK, M. How to design the fair experimental classifier evaluation. *Applied Soft Computing*, v. 104, 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494621001423>. 52

43 CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, Association for Computing Machinery, New York, NY, USA, v. 2, n. 3, maio 2011. ISSN 2157-6904. Disponível em: <https://doi.org/10.1145/1961189.1961199>. 53

44 BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>. 53

45 BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 18, n. 9, p. 509–517, set. 1975. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/361002.361007>. 53

46 OMOHUNDRO, S. M. *Five Balltree Construction Algorithms*. [S.l.], 1989. 53

47 FISHER, R. *The design of experiments*. Edinburgh: Oliver and Boyd, 1935. 54

48 KOUIROUKIDIS, N.; EVANGELIDIS, G. The effects of dimensionality curse in high dimensional knn search. In: *2011 15th Panhellenic Conference on Informatics*. [S.l.: s.n.], 2011. p. 41–45. 66

49 DUNHAM, J. G.; SUN, M.-T.; TSENG, J. C. R. Classifying file type of stream ciphers in depth using neural networks. In: *The 3rd ACS/IEEE International Conference onComputer Systems and Applications*. IEEE, 2005. p. 97. Disponível em: <https://ieeexplore.ieee.org/document/1387088>. 78

50 ZHAO, Z.; ZHAO, Y.; LIU, F. Research on grain-128's cryptosystem recognition. In: *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference(IAEAC 2018)*. [S.l.: s.n.], 2018. 78, 79, 80, 81, 82

51 KHADIVI, P.; MOMTAZPOUR, M. Application of data mining in cryptanalysis. In: IEEE. *9th International Symposium on Communications and Information Technology*. 2009. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5341228>. 78

52 CHOU, J.-W.; LIN, S.-D.; CHENG, C.-M. On the effectiveness of using state-of-the-art machine learning techniques to launch cryptographic distinguishing attacks. In: *5th ACM workshop on Security and artificial intelligence*. [S.l.: s.n.], 2012. 78

53 PAN, J. Encryption scheme classification : a deep learning approach. *International Journal of Electronic Security and Digital Forensics*, v. 9, p. 381–395, 2017. Disponível em: <https://doi.org/10.1504/IJESDF.2017.087397>. 78, 79

54 ZHANG, W.; ZHAO, Y.; FAN, S. Cryptosystem identification scheme based on ascii code statistics. *Security and Communication Networks*, v. 2020, p. 10, 2020. Disponível em: <https://doi.org/10.1155/2020/8875864>. 78, 79, 80, 81

55 CARVALHO, L. A. V. de; SOUZA, W. A. R. D.; XEXEO, J. A. M. Agrupar textos cifrados é equivalente a agrupar textos legíveis. In: *Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*. [s.n.], 2009. Disponível em: <http://www.nilc.icmc.usp.br/til/stil2009_English/Proceedings/stil/Souza-57776_1.pdf>. 78, 79

56 LIMA, A. P. de; GOMES, P. R.; XEXEO, J. A. M. Novas formas de ataques de distinção a cifradores: Caminhos para o futuro. *Revista Militar de Ciência e Tecnologia*, v. 2º Quadrimestre, p. 12–17, 2009. Disponível em: <http://www.rmct.ime.eb.br/arquivos/RMCT_2_quad_2009/novas_formas_ataque.pdf>. 79

57 SHARIF, S. O.; KUNCHEVA, L.; MANSOOR, S. Classifying encryption algorithms using pattern recognition techniques. In: *IEEE International Conference on Information Theory and Information Security*. [s.n.], 2010. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5689769>. 79, 81

58 SOUZA, W. A. R. D.; TOMLINSON, A. A distinguishing attack with a neural network. In: *13th International Conference on Data Mining Workshops*. IEEE, 2013. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6753915>. 79, 80, 81

59 SWAPNA, S.; DILEEP, A. D.; SEKHAR, C. C.; KANT, S. Block cipher identification using support vector classification and regression. *Journal of Discrete Mathematical Sciences and Cryptography*, Taylor Francis, v. 13, p. 305–318, 2010. Disponível em: <https://doi.org/10.1080/09720529.2010.10698296>. 79, 80, 81

60 TAN, C.; JI, Q. An approach to identifying cryptographic algorithm from ciphertext. In: *8th IEEE International Conference on Communication Software and Networks*. [s.n.], 2016. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7586649>. 79, 80, 81

61 TAN, C.; LI, Y.; YAO, S. Novel identification approach to encryption mode of block cipher. *Advances in Intelligent Systems Research*, v. 16, p. 586–591, 2016. Disponível em: <https://www.atlantis-press.com/proceedings/icsma-16/25866870>. 79, 80, 81

62 TAN, C.; DENG, X.; ZHANG, L. Identification of block ciphers under cbc mode. *Procedia Computer Science*, v. 131, p. 65–71, 2018. Disponível em: <https://www.sciencedirect.com/journal/procedia-computer-science/vol/131/suppl/C>. 79, 80, 81

63 FAN, S.; ZHAO, Y. Analysis of cryptosystem recognition scheme based on euclidean distance feature extraction in three machine learning classifiers. *Journal of Physics: Conference Series*, v. 1134, 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1314/1/012184/meta>. 79, 80, 81

64 KHADIVI, P.; MOMTAZPOUR, M. Cipher-text classification with data mining. In: IEEE. *4th International Symposium on Advanced Networks and Telecommunication Systems*. [S.l.], 2010. 79

65 HU, X.; ZHAO, Y. Block ciphers classification based on random forest. *Journal of Physics: Conference Series*, v. 1168, 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1168/3/032015/meta>. 79, 80, 81

66 SHARIF, S. O.; MANSOOR, S. P. Performance evaluation of classifiers used for identification of encryption algorithms. *ACEEE Int. J. on Network Security*, v. 2, p. 42–45, 2011. 79, 81

67 TORRES, R. H.; OLIVEIRA, G. A. de; XEXEO, J. A. M.; SOUZA, W. A. R. D.; LIDEN, R. Identification of keys and cryptographic algorithms using genetic algorithm and graph theory. *IEEE Latin America Transactions*, v. 9, p. 178–183, 2011. Disponível em: <https://ieeexplore.ieee.org/abstract/document/5765571>. 79, 80, 81

68 OLIVEIRA, G. A. de; XEXEO, J. A. M. A aplicação de algoritmos genéticos no reconhecimento de padrões criptográficos. *Revista Militar de Ciência e Tecnologia*, v. 3º Trimestre, p. 3–15, 2013. Disponível em: <http://rmct.ime.eb.br/arquivos/RMCT_3_tri_2013/RMCT_025_E8A_11.pdf>. 79, 80, 81

69 RAY, P. K.; KANT, S.; ROY, B. K.; BASU, A. Classification of encryption algorithms using fisher's discriminant analysis. *Defence Science Journal*, v. 67, p. 59–65, 2017. 79, 80

70 PRADEEPTHI, K. V.; TIWARI, V.; SAXENA, A. Machine learning approach for analysing encrypted data. In: *2018 Tenth International Conference on Advanced Computing (ICoAC)*. IEEE, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8939101>. 79, 81

71 POPPELE, A.; EICHLER, R.; JäGER, R. Classification of cryptographic libraries. University of Stuttgart, 2017. Disponível em: <http://dx.doi.org/10.18419/opus-10309>. 83

72 LARA, P. C. da S. *Implementação de uma API para criptografia assimétrica baseada em curvas elípticas*. Dissertação (Mestrado) — Instituto Superior de Tecnologia em Ciências de Computação de Petrópolis, 2008. 83

ANEXO A – ALGORITMOS CRIPTOGRÁFICOS X ALGORITMOS DE ML

As tabelas a seguir relacionam os autores identificados com os algoritmos criptográficos e algoritmos de aprendizado de máquina utilizados. Pela análise das tabelas 58, 59, 60, observa-se que a maior quantidade de trabalhos aborda as cifras de bloco. Apenas os trabalhos de Dunham, Sun e Tseng(49) e Zhao, Zhao e Liu(50) abordaram o problema para o grupo das cifras de fluxo, conforme identificado na tabela 61. No trabalho (49), não foi identificado o tipo de cifra de fluxo utilizada, portanto não foi identificado na tabela. No entanto, utilizou-se de Redes neurais para identificar o tipo de arquivo.

Também foram estudadas uma ampla gama de tipos diferentes de informações transformadas em criptogramas. Foram identificados trabalhos que estudaram criptogramas gerados por diferentes fontes:

- Imagens e áudio: Nos trabalhos de Khadivi e Momtazpour(51), Chou, Lin e Cheng(52), Barbosa et al.(19), Pan(53), Zhao, Zhao e Liu(50), Zhang, Zhao e Fan(54).
- Textos em diversos idiomas: Carvalho, Souza e Xexeo(55), Mello e Xexeo(17)

Criptografia Simétrica de Bloco				
Algoritmo	DES	3DES	AES	IDEA
Redes Neurais	(14) (56) (55) (15) (57) (18) (19) (20) (17)		(14) (56) (55) (15) (57) (58) (20) (17)	(57)
SVM	(9) (59) (57) (60) (61) (62) (63)	(9) (59) (60) (61) (62) (63)	(9) (64) (59) (57) (60) (61) (62) (63)	(57) (63)
Árvore de Decisão	(57) (16), (17) (20) (18), (19)	(16)	(57) (16) (17) (20)	(57) (16)
Random Forest	(50) (63) (65) (54)	(63) (65) (54)	(50) (63) (65) (54)	(50) (63) (65) (54)
Naive Bayes	(57) (66) (18) (19) (20)		(57), (66) (20)	(57), (66)
Algoritmos Genéticos	(15)		(15) (67) (68)	
Grafos	(15)		(15) (67)	
KNN	(9)	(9)	(9) (64)	
Regressão Logística	(63)	(63)	(63)	(63)
Bagging	(57)		(57)	(57)
Adaboosting	(57)		(57)	(57)
Rotation Forest	(57)		(57)	(57)
Instance Based Learner	(57)		(57)	(57)
CNN	(53)		(53)	
Fisher Discrininant Analysis	(69)			
Linear Discriminant Analysis			(64)	
Expectation-Maximization	(70)			

Tabela 58 – Classificadores x Algoritmos Criptográficos Simétricos - Parte 1

Criptografia Simétrica de Bloco				
Algoritmo	BLOWFISH	TWOFISH	MARS	SERPENT
Redes Neurais	(18) (19) (20) (17)	(58) (20) (17)	(58)	(58) (20) (17)
SVM	(9) (59) (60) (61) (62) (63)			
Árvore de Decisão	(16) (17) (20) (18) (19)	(17) (20)		(17) (20)
Random Forest	(50) (63) (65) (54)			
Naive Bayes	(18) (19) (20)	(20)		(20)
Algoritmos Genéticos		(67) (68)	(67) (68)	(67) (68)
Grafos		(67)	(67)	(67)
KNN	(9)			
Regressão Logística	(63)			
Bagging				
Adaboosting				
Rotation Forest				
Instance Based Learner				
CNN				
Fisher Discrininant Analysis			(69)	
Linear Discriminant Analysis				
Expectation-Maximization				

Tabela 59 – Classificadores x Algoritmos Criptográficos Simétricos - Parte 2

Criptografia Simétrica de Bloco				
Algoritmo	CAMELLIA	RC	SMS4	L-block
Redes Neurais		(57) (18) (19) (20) (17) (58)		
SVM	(63)	(57) (9) (59) (60) (61) (62)	(63)	
Árvore de Decisão		(57) (16) (17) (20) (18) (19)		
Random Forest	(50) (63) (65) (54)	(50)	(50) (63) (65) (54)	
Naive Bayes		(57) (66) (18) (19) (20)		
Algoritmos Genéticos		(67) (68)		
Grafos		(67)		
KNN		(9)		
Regressão Logística	(63)			
Bagging		(57)		
Adaboosting		(57)		
Rotation Forest		(57)		
Instance Based Learner		(57)		
CNN				
Fisher Discrininant Analysis				
Linear Discriminant Analysis				
Expectation-Maximization				(70)

Tabela 60 – Classificadores x Algoritmos Criptográficos Simétricos - Parte 3

Criptografia Simétrica de Fluxo				
Algoritmo	SALSA	TRIVIUM	DRAGON	SOSEMANUK
Redes Neurais				
SVM				
Árvore de Decisão				
Random Forest	(50)	(50)	(50)	(50)
Naive Bayes				
Algoritmos Genéticos				
Grafos				
KNN				
Regressão Logística				
Bagging				
Adaboosting				
Rotation Forest				
Instance Based Learner				
CNN				
Fisher Discrininant Analysis				
Linear Discriminant Analysis				
Expectation-Maximization				

Tabela 61 – Classificadores x Algoritmos Criptográficos Simétricos de Fluxo

ANEXO B – LINGUAGEM DE PROGRAMAÇÃO E BIBLIOTECAS UTILIZADAS

Os algoritmos para execução dos experimentos foram desenvolvidos na linguagem *python* e foi utilizada a biblioteca *scikit-learn* para os algoritmos de aprendizado de máquina. A escolha por esta biblioteca é justificada pela mesma ser a referência neste campo.

Poppele, Eichler e Jäger(71) realizaram um estudo sobre as bibliotecas criptográficas em diversas linguagens. Dentre as bibliotecas analisadas para a linguagem python, foi adotada nesta pesquisa a biblioteca Pycrypto, por possibilitar encriptar nos algoritmos RSA e Elgamal de forma mais próxima à teoria, sem algoritmos complementares de segurança.

Para a encriptação em logaritmos discretos em curvas elípticas, foi utilizada a biblioteca "*Fastecdsa*" apenas para a realização das operações de multiplicação de números escalares por pontos. Esta foi escolhida pelo seu desempenho, em termos de velocidade de computação das operações. A modelagem do algoritmo de logaritmo discreto em curvas elípticas seguiu o modelo explicado por Lara(72) no capítulo 2.

Os códigos desenvolvidos e arquivos utilizados estão disponíveis em: https://github.com /danielcapello/mestrado.