2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI) | 979-8-3503-7336-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/EICEEAI60672.2023.10590185

Cryptographic Algorithm Identification through Machine Learning for Enhanced Data Security

1st Ali Rachini
Department of Computer Science and Information Technology
Holy Spirit University of Kaslik (USEK)
Jounieh, Lebanon
alirachini@usek.edu.lb

2nd Maroun Abi Assaf
Department of Computer Science and Information Technology
Holy Spirit University of Kaslik (USEK)
Jounieh, Lebanon
maroun.abiassaf@usek.edu.lb

3rd Charbel Fares
Department of Computer Science and Information Technology
Holy Spirit University of Kaslik (USEK)
Jounieh, Lebanon
charbelfares@usek.edu.lb

4th Rida Khatoun
Telecom ParisTech
Paris, France
rida.khatoun@telecom-paristech.fr

***Abstract*—This paper discusses the significance of identifying encryption algorithms in today's digital era to ensure data security. The study uses machine learning (ML) techniques, including Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (KNN), to develop a classification model for distinguishing between encryption algorithms like Blowfish, AES, and 3DES. Results show distinct performances among the algorithms, with SVM achieving a robust 91% accuracy rate, Random Forest excelling in precision with a 99% accuracy rate, and KNN providing reasonable but comparatively lower accuracy at 34%. The findings underscore the diverse capabilities of ML algorithms in encryption algorithm identification, offering valuable insights for enhancing data security practices and emphasizing the importance of selecting the most suitable ML approach based on specific security requirements.**

***Index Terms*—Machine Learning, Encryption Algorithm Detection, Symmetric Algorithm, Data Security.**

I. Introduction

In the contemporary digital landscape, the paramount concern is safeguarding data integrity and confidentiality, emphasizing the critical role played by encryption algorithms. These algorithms, such as AES (Advanced Encryption Standard), 3DES (Triple Data Encryption Standard), and Blowfish, represent indispensable tools for transforming sensitive information into an indecipherable format. These cryptographic methods offer varying levels of security and computational efficiency. However, the ability to discern and categorize the encryption algorithm applied to encrypted text is pivotal in the evaluation of data security and potential vulnerabilities [1].

Machine learning (ML) techniques have exhibited remarkable prowess across diverse domains, encompassing natural language processing and image recognition‘ [2]. Within the realm of data security, ML offers a promising avenue for refining the classification of encrypted text [3]. By harnessing ML algorithms, it becomes feasible to scrutinize and uncover patterns within encrypted data, thereby facilitating the identification of the underlying encryption algorithm in use.

This Paper undertakes the exploration of an innovative approach employing ML for the classification of encrypted text under the auspices of the AES, 3DES, and Blowfish encryption algorithms [4]. The primary objectives are twofold: firstly, to construct an ML-based classification model endowed with the capability to accurately determine the encryption algorithm employed in encrypted text. Secondly, to delve into the repercussions of encryption algorithm detection on data security. By fulfilling these objectives, this research endeavors to illuminate the efficacy of ML techniques in fortifying the security of encrypted data, thereby imparting invaluable insights into the symbiotic relationship between ML and data security. The outcomes of this study hold the potential to guide the development of more robust encryption algorithms and classification methodologies, thus contributing to the advancement of data security practices.

Within the domain of data security, the capacity to categorize encrypted text plays an instrumental role in ascertaining the encryption algorithm in use and gauging its security level. Conventional methods of classification often fall short in accurately identifying the encryption algorithm due to the intricate and non-linear nature of encrypted data. Consequently, there exists a compelling necessity for a novel approach that leverages ML techniques to proficiently classify encrypted text. This research endeavors to confront this

challenge by proposing an innovative ML-based approach for the effective classification of encrypted text under the aegis of AES, 3DES, and Blowfish encryption algorithms [5].

The remaining of this paper is structured as follows: In Section 2, we conduct a comprehensive review of related research in the field. Section 3 is dedicated to elucidating the methodology used and the datasets harnessed for the purposes of this study. Subsequently, in Section 4, we expound upon the results garnered from the research endeavor. Section 5 serves as the platform for in-depth discussions pertaining to the study's findings, culminating in a comprehensive summation of the discoveries and contributions in the concluding remarks.

II. Related works

In our research, we aim to provide an overview of significant contributions in the field of cryptographic algorithm recognition. The paper [6] proposes encrypting weight data in neural networks using DES, AES, and Homomorphic Encryption to enhance data security in response to evolving threats. While AES provides a balanced solution, Fernet and HE offer privacy-centric alternatives, with performance considerations varying based on file size and computational resources. Authors in [7] introduce deep learning-based cryptanalysis on round-reduced SPECK, utilizing an "all-in-one" differential cryptanalysis approach. Inspired by this, the research demonstrates the use of ML to simulate all-in-one differentials for non-Markov ciphers, presenting distinguishers for various high-profile ciphers with reduced complexity.

Authors in [8] introduced a cryptographic recognition scheme employing support vector machines for the identification of five cryptographic regimes, namely AES, DES, 3DES, RC5, and Blowfish, with a focus on document classification techniques. Huang Liangdao et al. [9] put forth a hierarchical recognition scheme based on random forests, expanding the horizons of algorithm recognition problem research. Wu Yang et al. [10] innovatively merged ML and statistical techniques, employing a randomness test approach for ciphertext feature extraction. They further utilized the K-mean algorithm to construct a learner for the two-by-two recognition of five cryptographic algorithms, including AES, DES, SMS4, Camellia, and 3DES.

In [11], authors delved into cryptographic algorithm recognition under ciphertext-only conditions, utilizing feature extraction techniques like entropy calculation and Fourier transform. They employed the integrated learning hair method to undertake the recognition task for 11 contemporary cryptographic algorithms. Zhao Zhicheng et al. [12] introduced novel ciphertext features by employing 15 randomness tests from the NIST test set. They leveraged the random forest algorithm to successfully identify six grouped ciphers. Ji Wentao et al. [13] focused on the recognition of the national cipher SM4 algorithm and other grouped algorithms, based on randomness tests, ultimately achieving a high level of recognition accuracy. These notable endeavors collectively contribute to the evolving landscape of cryptographic algorithm recognition within the realm of data security and cryptography research.

In the realm of cryptographic algorithms, Kahate [14] asserts that one algorithm, in particular, held the spotlight as the most widely utilized for over two decades. However, its popularity waned due to the emergence of vulnerabilities. Addressing these concerns, Tanenbaum [15] posits that while the original algorithm exhibits security shortcomings, strategic upgrades can render it a viable option. Pfleeger and Pfleeger [16] underline the potential to bolster its security by employing a sequence of substitution and transposition techniques.

In response to the inherent vulnerabilities of the Data Encryption Standard (DES), the Blowfish algorithm emerged as a prospective alternative [17]. This transition spurred comparative analyses between DES and Blowfish. Noteworthy contributions include the study by Nie, Song, and Zhi [18], which delves into the comparative evaluation of speed and energy consumption, and the research by Verma, Agarwal, Dafouti, and Tyagi [19], which elucidates that Blowfish surpasses DES, AES, and Triple DES not only in terms of speed but also in the security enhancements it offers, primarily owing to its key size. Poonia and Yavad [20] further demonstrate the potential for modifications to enhance the algorithm's compactness and security compared to its original iteration.

In the context of stream cipher security software, RC4, also known as ARC4, plays a pivotal role, featuring prominently in applications like TLS, SSL, and WEP [21]. Despite its simplicity, efficiency, and speed, being five times faster than DES [22], it grapples with several exploitable weaknesses [23], [24]. According to Vanhoef and Piessens [25], the usage of RC4 is now discouraged due to its vulnerabilities.

Turning to asymmetric encryption, the RSA algorithm stands as a prominent figure [26]. It represents the inaugural publication in the realm of asymmetric algorithms [27] and hinges its security on the formidable challenge of factoring large prime numbers. Coutinho [26] emphasizes the critical importance of judiciously selecting these prime numbers, as improper choices can undermine the encryption's robustness. Despite these challenges, RSA finds applications in encoding and decoding medical images [28] and serves as the foundation for a hybrid Bluetooth communication algorithm [29].

Data mining, an integral process for extracting meaningful patterns from extensive datasets, is underpinned by various algorithms [30], [31]. The J48 classifier, an embodiment of the classic C4.5 decision tree data mining algorithm, offers two

pruning methods to reduce time complexity [32]. In parallel, the Multilayer Perceptron, a well-recognized neural network classifier [33], comprises an input layer, intermediate layers, and an output layer. The training phase is conducted under supervision, employing backpropagation to minimize error, thereby contributing to its efficacy in classification tasks.

III. Methodology

In this section, we will illustrate the data pre-processing and dataset generation steps within the context of our system model.

A. System Model

In our system model, Fig.1, the process begins with the generation of a text file comprising entirely of randomly selected letters. This file is structured to contain a total of 50,000 lines, each line consisting of 1,000 characters. Subsequently, we initiate the encryption phase where each line in the text file undergoes encryption using three distinct algorithms: Blowfish, AES, and 3DES. Notably, for AES and Blowfish, a key size of 128 bits is employed, while 3DES operates with a key size of 168 bits. Following this encryption process, we have established the foundation of our dataset.

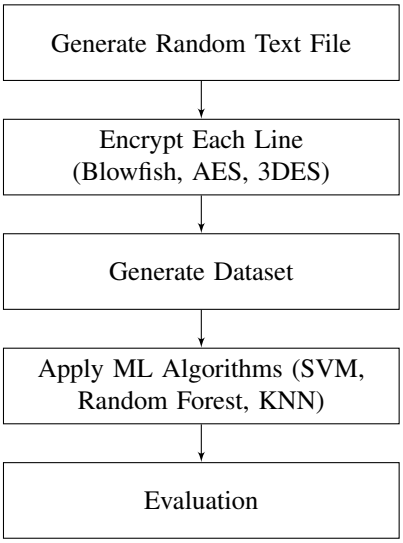

Fig. 1. System Model Diagram

With the dataset in place, we proceed to apply a set of ML algorithms for training and testing. The selected algorithms for this purpose include Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (KNN). These algorithms are instrumental in analyzing and making predictions based on the encrypted data, thereby contributing to the evaluation and enhancement of our system's performance.

B. Data Preprocessing and splitting

For data preprocessing, our dataset is already well-structured. There are no significant data gaps as we generated the dataset intentionally, resulting in the absence of duplicated or null values. Moreover, the dataset is inherently balanced. The primary data preprocessing task involves transforming categorical data into numerical values. For instance, we assign numerical values to categories, such as representing Blowfish as 0, AES as 1, and 3DES as 2.

Following data preprocessing, the next step involves splitting the dataset into two subsets: 80% for training and 20% for testing. This division enables us to train our ML algorithms on a significant portion of the data while reserving a separate portion for evaluating their performance.

Fig.2 provides a visual representation of the dataset, showcasing the distribution of encrypted data across different encryption algorithms.

0	bvXU3VcDUiTPyY6RoJPrfk8kd+yGp+y5Sd/wamWWH4gU+l...	0
1	4x13hsryQkYYgoXR4PHxqwKPrnMMxDEoNHYFO1BX7jkxye...	1
2	V3btmx1VgK88ZtjRCyPzp2dPWcVOz364YcsZXOsgdjQjli...	2
3	zT60nLR0OC+gDvloThTrPrQ4KwxbPEpS7Sh3jD1+mh/fLg...	0
4	aJOHja+Zdls7umLepV6lfclYHDf7eNJ6Zy3P4RipA9KlJg...	1

Fig. 2. Encrypted Data Distribution Across Algorithms

The first column serves as an index, providing a unique identifier for each entry in the dataset. The second column is dedicated to the encrypted data itself, representing the ciphertext resulting from the application of encryption methods. The third column classifies the data by indicating the specific encryption algorithms used; each entry in this column designates the type of algorithm employed for the encryption process.

IV. Simulation Results

A. Evaluation metrics

1) Confusion Matrix: A confusion matrix in ML is a tabular representation used to assess the performance of a classification algorithm, particularly in binary classification tasks. It classifies model predictions into four categories: True Positives (TP - correctly predicted positives), True Negatives (TN - correctly predicted negatives), False Positives (FP - incorrectly predicted positives), and False Negatives (FN - incorrectly predicted negatives). These parameters provide crucial insights into a model's ability to correctly classify instances and are fundamental for calculating various performance metrics like accuracy, precision, recall, and the F1 score, aiding in the evaluation and improvement of the model's predictive capabilities [34].

2) Precision: Precision is a metric that quantifies the proportion of true positive predictions [34]. It is defined by the equation:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

3) Recall: Recall, often referred to as the true positive rate, measures the ability of a model to correctly identify positive instances [34]. Its equation is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

4) F1-Score: The F1-Score is a harmonic mean of precision and recall, offering a balanced evaluation metric. It can be calculated using the following equation [34]:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

5) Classification Report: The classification report provides a summary of model performance for each class in a multi-class classification problem. It includes metrics like precision, recall, F1-Score, and support, presented in a tabular format.

6) Mean Squared Error (MSE): Mean Squared Error (MSE) is a fundamental metric widely used in ML and statistical analysis to assess the accuracy and performance of predictive models, particularly in regression tasks. MSE measures the average of the squared differences between the predicted values and the actual or observed values. It is a crucial tool for quantifying the quality of predictions and understanding the extent of errors in a model's output.

The MSE is calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y_i})^2 \tag{4}$$

Where:

- n represents the total number of data points or observations.
- Y_i denotes the actual or observed value for the ith data point.
- $\hat{Y_i}$ represents the predicted value for the ith data point.

A lower MSE value indicates that the model's predictions are closer to the actual values, signifying a better fit. Conversely, a higher MSE suggests that the model's predictions deviate significantly from the actual data, indicating poorer performance. The square operation in the MSE formula ensures that both overestimations and underestimations contribute to the error score.

B. Proposed Algorithms

In this section, we introduce and discuss the algorithms proposed for our research, which are instrumental in classifying encrypted data according to their respective encryption methods. The algorithms presented here are designed to enhance the accuracy and efficiency of encryption algorithm recognition.

1) Support Vector Machine (SVM): Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification tasks. In our research, SVM is applied to recognize the encryption algorithm used based on the features extracted from encrypted data. SVM aims to find the optimal hyperplane that best separates data points belonging to different encryption algorithms while maximizing the margin between classes.

The decision function for SVM is given by:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n}\alpha_i y_i K(x, x_i) + b\right) \tag{5}$$

Where:

- $f(x)$ represents the predicted class label.
- α_i are the Lagrange multipliers.
- y_i is the class label of the training sample.
- $K(x, x_i)$ is the kernel function.
- b is the bias term.

2) Random Forest: Random Forest is an ensemble learning method that leverages multiple decision trees to make predictions. In our research, Random Forest is employed for encryption algorithm recognition. It combines the results of several decision trees to achieve robust and accurate classification.

Random Forest combines the predictions of N decision trees to obtain the final prediction:

$$\hat{Y}_{\text{RF}} = \frac{1}{N}\sum_{i=1}^{N}\hat{Y}_i \tag{6}$$

Where:

- $\hat{Y}_{\text{RF}}$ is the Random Forest prediction.
- $\hat{Y}_i$ represents the prediction of the ith decision tree.

3) k-Nearest Neighbors (KNN): k-Nearest Neighbors is a simple yet effective classification algorithm. It assigns a class label to an encrypted data point based on the majority class of its k-nearest neighbors in the feature space. In our research, KNN is used to identify the encryption algorithm by analyzing the similarities between the encrypted data points.

The predicted class label using KNN is based on the majority class among the k nearest neighbors:

$$\hat{Y} = \text{majority}\left(\{y_i\}_{i \in \mathcal{N}_k(x)}\right) \tag{7}$$

Where:

- $\hat{Y}$ is the predicted class label.
- y_i represents the class labels of the k nearest neighbors of data point x.

These proposed algorithms are pivotal in achieving our research objectives, which include accurately classifying encrypted data according to the encryption methods used. Each algorithm brings its unique strengths to the task of encryption algorithm recognition, and their combined application allows us to explore the effectiveness of different ML techniques in enhancing data security practices.

C. Results

Table I provides an overview of the results obtained from a classification task using three different ML algorithms: Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (KNN). The classification task involved categorizing data into three classes representing different encryption algorithms: Blowfish, AES, and 3DES.

TABLE I
CLASSIFICATION REPORT FOR MACHINE LEARNING ALGORITHMS

Algorithm	Class	Precision	Recall	F1-Score	Accuracy	Mean Squared Error
SVM	Blowfish AES 3DES	0.90 0.91 0.92	0.97 0.81 0.95	0.93 0.86 0.94	91%	0.013
Random Forest	Blowfish AES 3DES	0.99 1.00 0.97	1.00 0.96 1.00	0.99 0.98 0.99	99%	0.089
KNN	Blowfish AES DES3	0.34 0.35 0.35	0.46 0.25 0.32	0.39 0.29 0.33	34%	1.404

The SVM exhibited a strong performance across all encryption classes. For Blowfish, it achieved a high precision of 0.90, indicating that it correctly identified Blowfish instances with a 90% accuracy. The recall for Blowfish was 0.97, signifying that it captured 97% of the actual Blowfish instances. The F1-Score for Blowfish was 0.93, which demonstrates a balanced performance in terms of precision and recall. SVM also performed well for AES and 3DES, with precision values of 0.91 and 0.92, respectively. The accuracy of SVM across all classes was 91%, highlighting its effectiveness. The Mean Squared Error (MSE) was low, indicating a small difference between predicted and actual values.

Random Forest excelled in precision for all classes, achieving values of 0.99 for Blowfish, 1.00 for AES, and 0.97 for 3DES. The recall and F1-Scores were also commendable, particularly for Blowfish. Random Forest's overall accuracy was the highest among the algorithms at 99%, indicating extremely accurate classification. The MSE was also low, suggesting precise predictions.

KNN's performance, while still reasonable, was relatively lower compared to the other two algorithms. For Blowfish, it exhibited a lower precision (0.34) and an F1-Score of 0.39, indicating a relatively higher rate of false positives. The accuracy for KNN was the lowest among the algorithms at 34%. The MSE was the highest, indicating larger differences between predicted and actual values.

The results demonstrate that Random Forest achieved the highest overall accuracy and F1-Scores across all classes, indicating its superior performance in this classification task. SVM also delivered strong results, particularly for Blowfish and 3DES. In contrast, KNN showed lower accuracy and F1-Scores, indicating a less accurate classification compared to the other two algorithms. The choice of the most suitable algorithm should be made based on the specific requirements of the classification task and the desired balance between precision and recall.

V. CONCLUSION

This study delved into the classification of encrypted data using Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (KNN) algorithms to discern encryption algorithms like Blowfish, AES, and 3DES. The findings revealed distinctive performances: SVM demonstrated robust accuracy (91%) and F1-Scores; Random Forest excelled with 99% accuracy, high precision, recall, and F1-Scores; KNN exhibited moderate accuracy (34%). Numerical breakdowns underscored algorithmic strengths and trade-offs, aiding practitioners in choosing models based on specific needs. This research contributes insights into ML for encryption algorithm identification, with potential implications for data security enhancement. Further exploration could refine classifications by investigating additional algorithms and feature engineering techniques.

REFERENCES

[1] H. Dibas and K. E. Sabri, "A comprehensive performance empirical study of the symmetric algorithms: Aes, 3des, blowfish and twofish," in *2021 International Conference on Information Technology (ICIT)*, pp. 344–349, IEEE, 2021.

[2] X. Zhang, F. T. Chan, and S. Mahadevan, "Explainable machine learning in image classification models: An uncertainty quantification perspective," *Knowledge-Based Systems*, vol. 243, p. 108418, 2022.

[3] G. Abbas, A. Mehmood, M. Carsten, G. Epiphaniou, and J. Lloret, "Safety, security and privacy in machine learning based internet of things," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, p. 38, 2022.

[4] S. M. Radhi and R. Ogla, "In-depth assessment of cryptographic algorithms namely des, 3des, aes, rsa, and blowfish," *IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING*, vol. 23, no. 3, pp. 125–138, 2023.

[5] K. Yuan, D. Yu, W. Yang, Z. Du, L. Shen, and Z. Li, "Identification of block cipher algorithms using multi-layer perception algorithm," 2023.

[6] S.-Q. Yeow and K.-W. Ng, "Neural network based data encryption: A comparison study among des, aes, and he techniques," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 3-2, 2023.

[7] A. Baksi, "Machine learning-assisted differential distinguishers for lightweight ciphers," in *Classical and Physical Security of Symmetric Key Cryptographic Algorithms*, Computer Architecture and Design Methodologies, Springer, Singapore, 2022.

[8] A. D. Dileep and C. C. Sekhar, "Identification of block ciphers using support vector machines," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 2696–2701, IEEE, 2006.

[9] H. Liangtao, Z. Zhicheng, and Z. Yaqun, "A two-stage cryptosystem recognition scheme based on random forest [j]," *Chinese Journal of Computers*, vol. 41, no. 2, pp. 382–399, 2018.

[10] Y. Wu, T. Wang, M. Xing, and J. Li, "Block ciphers identification scheme based on the distribution character of randomness test values of ciphertext," *Journal on Communications*, vol. 36, no. 4, pp. 146–155, 2015.

[11] L. Hongchao, "Research on cryptographic algorithm recognition based on ciphertext feature [d]," *Xidian University*, 2018.

[12] Z. Zhao, Y. Zhao, and F. Liu, "Research on grain-128's cryptosystem recognition," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2013–2017, IEEE, 2018.
[13] W. Ji, Y. Li, and B. Qin, "Identification of sm4 block cipher system based on random features," *Appl. Res. Comput*, vol. 50, 2021.
[14] A. Kahate, *Cryptography and Network Security*. Nova Delhi: McGraw Hill Education, 3rd ed., 2013.
[15] A. S. Tanenbaum, *Computer Networks*. Boston: Pearson, 5th ed., 2011.
[16] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*. Boston: Prentice Hall, 2006.
[17] B. Schneier, "Fast software encryption," in *Cambridge Security Workshop Proceedings*, pp. 191–204, 1994.
[18] T. Nie, C. Song, and X. Zhi, "Performance evaluation of des and blowfish algorithms," in *International Conference on Biomedical Engineering and Computer Science (ICBECS)*, (Wuhan), pp. 1–4, 2010.
[19] O. P. Verma, R. Agarwal, D. Dafouti, and S. Tyagi, "Performance analysis of data encryption algorithms," in *3rd International Conference on Electronics Computer Technology (ICECT)*, (Kanyakumari), pp. 399–403, 2011.
[20] V. Poonia and N. S. Yadav, "Analysis of modified blowfish algorithm in different cases with various parameters," in *International Conference on Advanced Computing and Communication Systems*, (Coimbatore), pp. 1–5, 2015.
[21] M. M. Hammood, K. Yoshigoe, and A. M. Sagheer, "Rc4-2s: Rc4 stream cipher with two state tables," *Information Technology Convergence*, vol. 253, pp. 13–20, 2013.
[22] S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B. Sinha, "High-performance hardware implementation for rc4 stream cipher," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 730–743, 2013.
[23] S. Fluhrer, I. Mantin, and A. Shamir, "Weakness in the key scheduling algorithm of rc4," in *Selected Areas in Cryptography*, vol. 2259, pp. 1–24, 2001.
[24] I. Mantin and A. Shamir, "A practical attack on broadcast rc4," in *Fast Software Encryption*, vol. 2355, pp. 152–164, 2002.
[25] M. Vanhoef and F. Piessens, "All your biases belong to us: Breaking rc4 in wpa-tkip and tls," in *Proceedings of the 24th USENIX Conference on Security Symposium*, (Washington), pp. 12–14, 2015.
[26] C. S. Coutinho, *Números Inteiros e Criptografia RSA*. 2003.
[27] A. K. Das and C. E. V. Madhavan, *Public-key Cryptography Theory and Practice*. Deli: Pearson, 2009.
[28] N. Anane, M. Anane, H. Bessalah, M. Issad, and K. Messaoudi, "Rsa based encryption decryption of medical images," in *7th International Multi-Conference on Systems Signals and Devices (SSD)*, pp. 1–4, 2010.
[29] W. Ren and Z. Miao, "A hybrid algorithm based on des and rsa in bluetooth communication," in *Second International Conference on Modeling, Simulation, and Visualization Methods (WMSVM)*, (Sanya), pp. 221–225, 2010.
[30] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Burlington: Morgan Kaufmann, 3rd ed., 2011.
[31] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Waltham: Morgan Kaufmann, 3rd ed., 2011.
[32] W. N. H. W. Mohamed, M. N. M. Sallen, and A. H. Mohamed, "A comparative study of reduced error pruning method in decision tree algorithms," in *IEEE International Conference of Control System, Computing and Engineering*, (Penang), pp. 23–25, 2012.
[33] L. N. C. Silva, "Análise e síntese de estratégias de aprendizado para redes neurais artificiais," Setembro 1998.
[34] D. Axman and R. Yacouby, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," 2020.