International Joint Conference on Industrial Engineering and Operations Management- ABEPRO-ADINGOR-IISE-AIM-ASEM (IJCIEOM 2020)

Information Security and Machine Learning: Encryption Allocation Based on Recognizing of Text Patterns

Santos P H J[1], Costa Junior C A D F[2], Rodrigues J[3], Carvalho L M[4], Souza F H B[5]

Abstract The productive processes have presented a high demand and generation of data and information. However, risks and attacks on the digital data of the companies have presented significant threats and losses to the industries that are embedded in the industrial revolution 4.0. Thus, this paper aims to present the feasibility study of an automated methodology, focused on cryptographic allocation, according to the degree of confidentiality of its content. Based on text pattern recognition algorithms such as Multilayer Perceptron (MLP) and Support Vector Machine (SVM), the proposed solution promotes a series of experiments in order to analyze whether certain information is confidential or "very" confidential. After analyzing the information within the message content, using text pattern recognition techniques, the encryption algorithm responsible for encoding the message is chosen. With this approach it possible to choose either stronger encryption for very confidential information or a weaker encryption for not-so-confidential information. The experimentation has a performance analysis in order to evaluate the computational cost of the processes involved.

Keywords: Machine Learning; Information Security; Cryptography; Classified Information.

1 Introduction

Efficient and secure communication has been, and still is, extremely important throughout the history of mankind, and it is not different today. With the expansion of the internet technology connecting the world, the communication and data traffic of confidential information has been growing in a proportion never seen before. Nowadays, it is possible to have important business and political meetings through digital media, industrial data traffic and storage through cloud computing with the advent of the 4.0 industry, and so on. Based on that, aggravating factors have been raised such as: how to intelligently ensure the security of the data transmitted? How to assign proper security to each data? Does such a technique have a high computational processing cost? (Pejic-Bach, et al., 2020; Khalifa, et al., 2004).

In this context, two main concepts are pertinent: Machine Learning (Pattern Recognition) and Cryptography. This work relies on the use of pattern recognition, which aims to construct a simpler representation of a data set through its most relevant characteristics. According to Jiang et. al. (2020), pattern recognition is the field of science that aims to classify objects into a certain number of categories or classes by observing their characteristics. One of the metrics that is used to evaluate the accuracy of a pattern recognition algorithm is the Receiver Operator Characteristic curve (ROC) and the Area Under the Curve (AUC) (Mazurowski & Tourassi, 2009). This analysis has been widely employed as a useful and

[1]Pedro Henrique Junio Santos (e-mail: pedro_ssantoss@yahoo.com.br).

[2]Carlos Augusto de Faria Costa Junior (e-mail: carlosjr4023@gmail.com).

[3]Diva de Souza e Silva Rodrigues (e-mail: divasouz@gmail.com)

[4]Luiz Melk de Carvalho (e-mail: luizmelk22@hotmail.com)

[5]Flávio Henrique Batista de Souza (e-mail: flabasouza@yahoo.com.br)

Centro Universitário de Belo Horizonte UNIBH. Av. Prof. Mário Werneck, 1685, Buritis, Belo Horizonte, Minas Gerais, Brazil

powerful tool in several Machine Learning, Data Mining and medical decision-making applications, in which a graphical method is used for evaluation, organization and selection of diagnostic systems (Fawcett, 2005).

Another relevant aspect of this research relies on the encryption, which can be considered as a set of methods and techniques for encrypting or encoding readable information, converting original text into unreadable text, and it is possible, through a reverse process, to retrieve the original information (Boyd & Mathuria, 2003).

Some works have already been presented with the same perspective, such as Souza et al. (2009), who developed different approaches for the evaluation of cryptography, to select the most suitable algorithm for the task, according to the sets of requirements to be evaluated. Within these requirements it is possible to use the pattern recognition generated from these algorithms. However, in the work of Souza et al. (2009), it was focused on several methods and algorithms to make various encryption for texts, without evaluating which algorithm would bring better performance and lower cost adequacy for each related text. Thus, unlike Souza et al. (2009), the purpose of the work is to analyze each text, adapting the proper encryption for this text to guarantee the best security and algorithm performance according to the needs of the analyzed text.

In the work of Oliveira et al. (2006), patterns were identified in cryptograms, according to DES (Data Encryption Standard) and AES (Advanced Encryption Standard) algorithms, acquiring the characterization of cryptograms with the same key. However, it is not yet known in which approach the encryption allocation should be based on, and how this will correlate to the performance of the developed framework. In addition, another fundamental point is related to the accuracy of the applied methodology.

Currently, several industrial sectors have invested in strategic data management technologies based on machine learning and pattern recognition, such as health, security and supply chain (Gan, et al., 2016; Li, et al., 2015; Lima-Junior & Carpinetti, 2019; Souza, et al., 2019).

Based on that, some questions arise such as: Which pattern recognition algorithm would have better performance with the text analysis process? What are the settings? How would this structure correlate to cryptographic algorithms? What would the proposed full structure look like? These questions will be developed throughout this paper.

2 Objectives

The main objective of this research is to propose and demonstrate an experimental analysis, based on a consolidated methodology, to indicate the type of encryption, according to the degree of confidentiality of the information. In this context, the specific objectives of this paper are: to evaluate a pattern recognition framework that will have the purpose to interpret the text through the level of confidentiality; describe a methodology of what will be the process of choosing the encryption algorithm according to the degree of confidentiality that will be presented; to evaluate the degree of accuracy of different pattern recognition algorithms and evaluate the efficiency of the entire process based on the computational cost of the encryption methods considered to be applied to the dataset.

This work is justified by the necessity to assign the best encryption according to the information confidentiality relevance, but with a conscious consumption of computational resources.

3 Methods

The research methodology followed two steps: the definition of the methods of the experiments to be performed and the definition of the dataset to be used.

3.1 Definition of Experiment Methods

The proposed framework has the following assumptions: A widely recognized Machine Learning-based pattern recognition algorithm must recognize the pattern of a “confidential” or “very confidential” message. Similar to the methodology developed by Souza et al. (2019), pattern recognition algorithms widely used

in the literature with different parameter configurations were used to explore the parameter configurations of the selected algorithms, in this case the Multilayer Perceptron (MLP) and Support Vector Machine (SVM) (Frias-Martinez, et al., 2006; Thome, 2012).
According to the recognized pattern, the framework recommends two possible encryptions (RSA or Cesar Cipher). In the experiments of this paper, the time and accuracy of pattern recognition are evaluated in order to present the implicit computational cost in the process.

3.2 Dataset for experiments

Although there are datasets that reach hundreds of thousands (even millions) of samples, with hundreds of words as variables; for this research, there was initially a process of choosing the dataset. It should contains: a considerable number of samples (between three and five thousand), so that an experimental structure that does not need robust computational resources for the recognition can be used; a number of approximately 50 words to be monitored (thus, in a company, the tool would give the user the option of adding a minimum number of words to be monitored for company security); the word incidence index was already normalized; and it had already been used in the literature.

Thus, a dataset, although old, that uses words that are still relevant in e-mails communication (in English), with the proposed range of samples and words is relevant for the proposed work. However, the experiment methodology is not limited to the proposed basis, and can be applied on any dataset that meets the scenario (number of samples and words). For the training of classified information, a real dataset generated between June and July 1999 was obtained from the Irvine Reeber repository, George Forman, Jaap Suermandt in conjunction with Hewlett-Packard.

This dataset has 2,788 normal email samples and 1,813 spam samples. The choice of using the dataset for training was mainly based on the labeling between these emails (spam or non-spam), which resembles the desired analysis for classification between sensitive or very sensitive information. In addition to the normalization of recorded data and having a relevant dictionary, which has 58 variables, where:

- 48 are attributes of type word_freq_WORD, continuous and real {0, ..., 100}: Corresponding to the percentage of words in the analyzed emails, that is correlated to the percentage of incidence of the word “WORD”, where the ratio between number of occurrences is made of the word “WORD” in the email. A word, in this case, is any string of alphanumeric characters limited by non-alphanumeric characters, or an end-of-string. Thus, 48 different words were referenced, such as CREDIT, INTERNET, WILL. Table 1 shows all 48 references;

Table 1. Word Frequency References (WORD_FREQ).

word_freq_make	word_freq_hp	word_freq_business
word_freq_address	word_freq_hpl	word_freq_email
word_freq_all	word_freq_geroge	word_freq_you
word_freq_3d	word_freq_650	word_freq_credit
word_freq_our	word_freq_lab	word_freq_your
word_freq_over	word_freq_labs	word_freq_font
word_freq_remove	word_freq_telnet	word_freq_000
word_freq_internet	word_freq_857	word_freq_money
word_freq_order	word_freq_data	word_freq_cs
word_freq_mail	word_freq_415	word_freq_meeting
word_freq_receive	word_freq_85	word_freq_original
word_freq_will	word_freq_technology	word_freq_project
word_freq_people	word_freq_1999	word_freq_re
word_freq_report	word_freq_parts	word_freq_edu
word_freq_addresses	word_freq_pm	word_freq_table
word_freq_free	word_freq_direct	word_freq_conference

- 6 are attributes of type char_freq_CHAR, continuous and real {0, ..., 100}: Percentage of characters in email, where the ratio between number of CHAR occurrences in the email and total number of characters in the email. Table 2 shows the reference characters;

Table 2. Character Frequency References (CHAR_FREQ).

char_freq_;	char_freq_[	char_freq_$
char_freq_(	char_freq_!	char_freq_#

- 1 capital_run_length_average, real and continuous attribute {1, ...}: average length of uninterrupted uppercase sequences;
- 1 attribute capital_run_length, longest, continuous and integer {1, ...}: length of the largest unbroken sequence of uppercase letters;
- 1 attribute capital_run_length_total, continuous and integer {1, ...}: sum of the length of uninterrupted capital letter sequences in the email;
- 1 nominal spam class attribute {0,1}: denotes whether the email was considered spam (1) or not (0), which means unsolicited commercial email.

In order to identify the best algorithm for neural network training, experiments were performed with MLP (Multilayer Perceptron) and SVM (Support Vector Machine) algorithms. The experiments were performed with the following hardware and software configurations: Operating System Windows 10 Home Single Language 64 Bit; Intel Core i5-7300 Processor - 2.5 GHz; 8GB RAM; video board NVIDIA GeForce GTX 1500.

4 Results

4.1 Text recognition pattern experiments

4.1.1 MLP experiments

The experiments performed for MLP and its variations (Backpropagation Standard, Backpropagation Momentum, Resilient Propagation, Backpropagation With Weight Decay, and Quick Propagation) were based on the number of hidden layer neurons, ranging from 2, 3, 4, 5, 7 and 10 neurons, and in the number of epochs, ranging from 100, 500, 1,000 and 5,000 epochs.

A total of 240 tests were performed for AUC and time measurement calculations. The graphs shown in Fig. 1 and 2 present the mean, minimum and maximum AUC and runtime values for MLP algorithms.

Analysing the indicators obtained in Fig. 1 and 2, it is observed that the Backpropagation Momentum algorithm (represented by BackpropMomentum) presented the best result compared to the other types (excluding outliers), since the maximum value of its AUC was 0.9451 with an average of 0.9282. In addition, it presented the second lowest average runtime of about 21.61 seconds.

Looking at the Backpropagation Standard results (represented by Std_Backpropagation), it is possible to notice that the algorithm also generated relevant results. The maximum value of its AUC was 0.9411, with an average AUC of 0.9275, and average execution time of 21.75 seconds. The results are close to those that were obtained by the Backpropagation Momentum variation. The Resilient Propagation (represented by RProp) yielded the best average AUC (0.9293), with a maximum AUC of 0.9427. However, it was the algorithm with the highest average time, with a value of 22.77 seconds.

The Backpropagation With Weight Decay algorithm (represented by BackpropWeightDecay) presented a maximum AUC of 0.9396 and an average of 0.9245. It is also observed that, for this type, the highest average execution time was achieved in relation to the other types, with a value of about 25.62 seconds.

The results generated by the Quick Propagation algorithm (represented by Quickprop) correspond to the lowest AUC of this experiment (disregarding outliers). It presented a maximum equivalent of 0.9337, with an average of 0.9167 and an average execution time of 20.19 seconds.

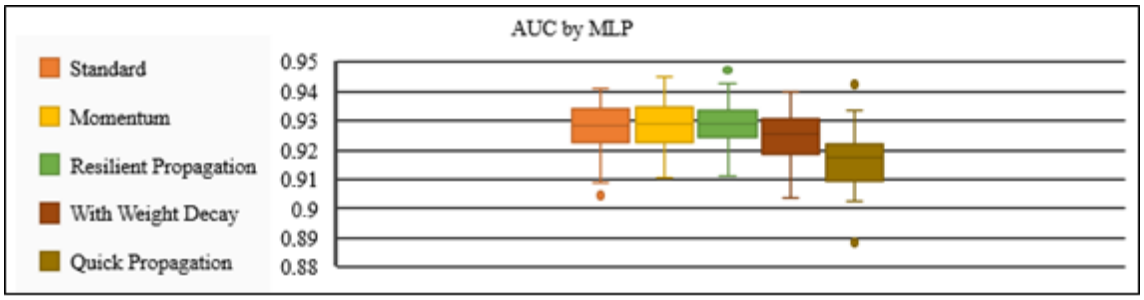

Fig. 1. AUC by MLP Type.

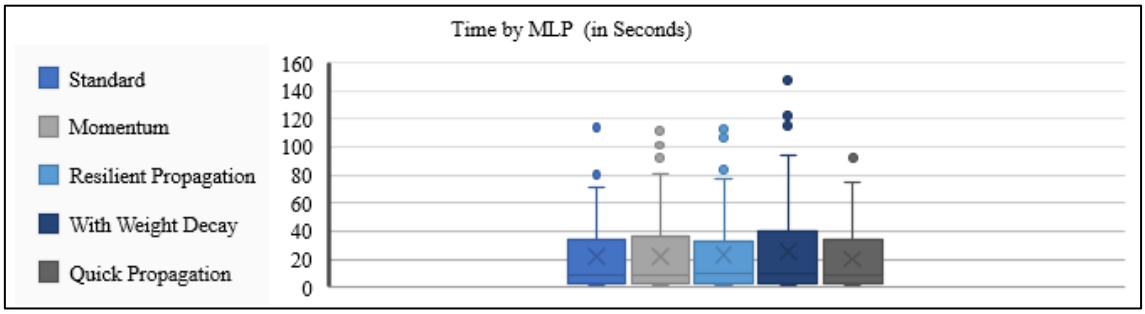

Fig. 2. Time by MLP Type.

4.1.2 SVM Experiments

The experiments performed for SVM were based on K-Fold Cross Validation, separated into subsets of k={3,5,7}, with breach costs ranging from 3, 5 and 7. Its kernel has variations in: Radial Gaussian, Polynomial Kernel, Linear Kernel, and Hyperbolic Tangent Kernel bases.

For AUC calculations and time measurement, a total of 36 tests were performed. Fig. 3 and 4 show the graphs of the average, minimum and maximum values of AUC and execution time obtained from the experiments performed with SVM algorithms.

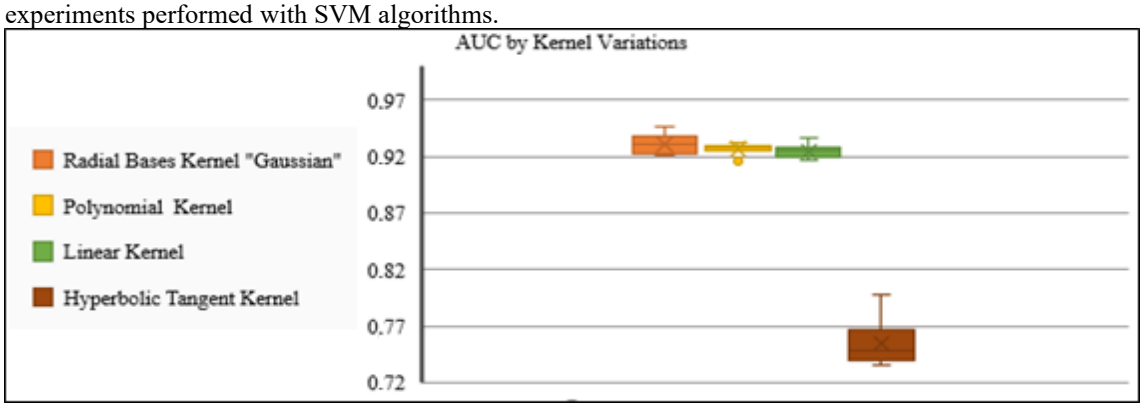

Fig. 3. AUC by Kernel Variations.

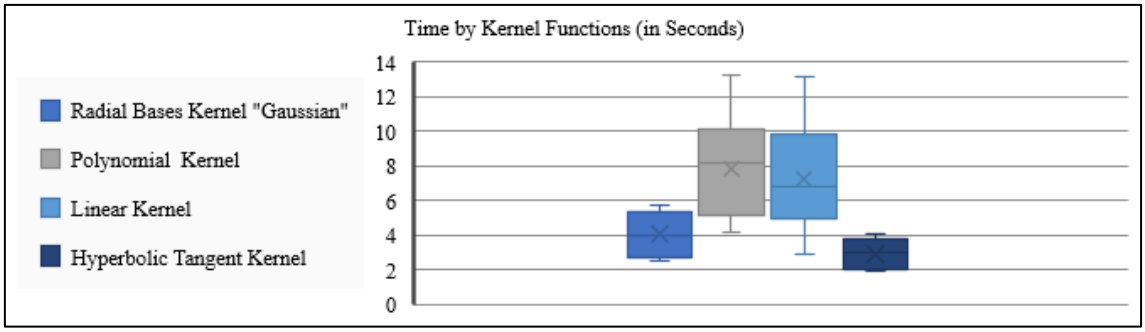

Fig. 4. Time by Kernel Variations.

The indicators in Fig. 3 and 4 show that the Gaussian Radial Bases Kernel (represented by rbfdot) presented a more significant result compared to other kernel variations, with a maximum AUC of 0.9457, an average AUC of 0.9210 and the average runtime of about 4.06 seconds. It is also important to notice that the Hyperbolic Tangent Kernel (represented by tanhdot) presented the lowest average runtime, of about 2.93 seconds. However, it is the process that achieved the smallest AUC of the four kernel variations tested, with a maximum of 0.7982 and an average of 0.7544.

With the results obtained by the Polynomial Kernel (represented by polydot), it is observed that this process achieved the highest average time compared to the others, about 7.82 seconds. Its maximum AUC resulted in 0.9318 with an average of 0.9268.

Looking at the results generated for the Linear Kernel (represented by vanilladot), the second largest recorded AUC is verified, with a peak of 0.9358 and an average of 0.9247. This process took, on average, 7.24 seconds per execution.

4.2 Encryption Experiments

In order to analyze the difference in execution time between cryptographic algorithms with symmetric and asymmetric methodology, a series of experiments were performed. It was used the Caesar Cipher method to represent the symmetrical model, and the RSA method for the asymmetric model.

The experiments were performed with the same hardware configurations used in the Text Pattern Recognition experiments. The results obtained from the experiments, as shown in Fig. 5 and 6, were based on the collection of 20 emails classified as confidential and very confidential.

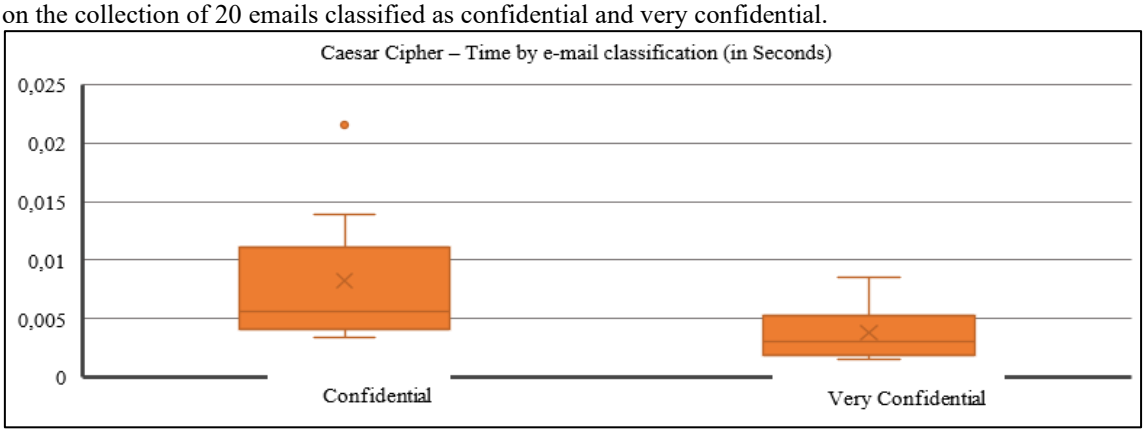

Fig. 5. Caesar Cipher – Time by e-mail classification.

A total of 40 tests were performed for average runtime calculations, with 20 runs performed for each selected encryption method. Analyzing the indicators of Fig. 5 and 6, it can be seen that, in both methods, the average execution time of the encryption algorithms was shorter for emails classified as very sensitive.

There is also a big difference in the average execution time between the methods. The Caesar's Cipher method had an average of 0.008 seconds for sensitive emails and an average of 0.003 seconds for very sensitive emails, while the RSA method got the longest execution times with an average of 7.81 seconds for emails considered to be sensitive and an average of 6.56 seconds for emails considered to be very sensitive.

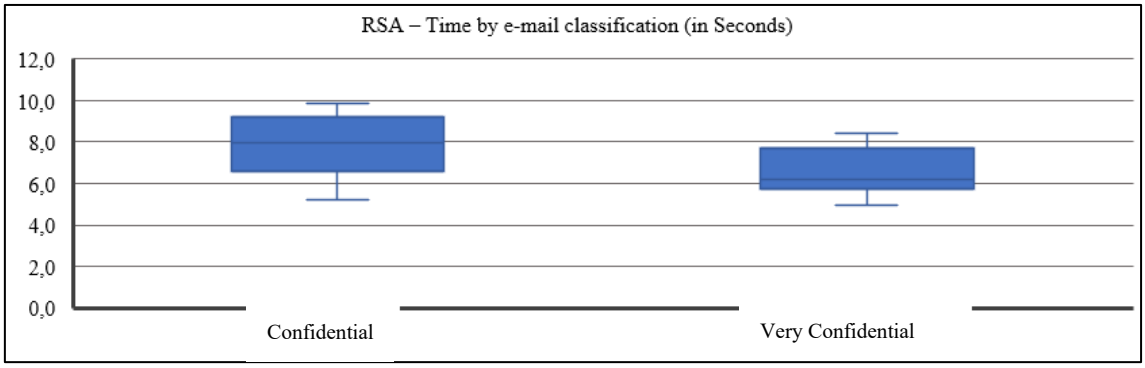

Fig. 6. RSA – Time by e-mail classification (in Seconds).

4.3 Resulting Structure

The structure elaborated for the proposed methodology (represented in Fig. 7) was organized as follows: In an email exchange, each forwarded email is subject to an evaluation in order to determine its degree of confidentiality through the algorithm. After the email content has been evaluated by the algorithm, it is labeled as confidential or very confidential.

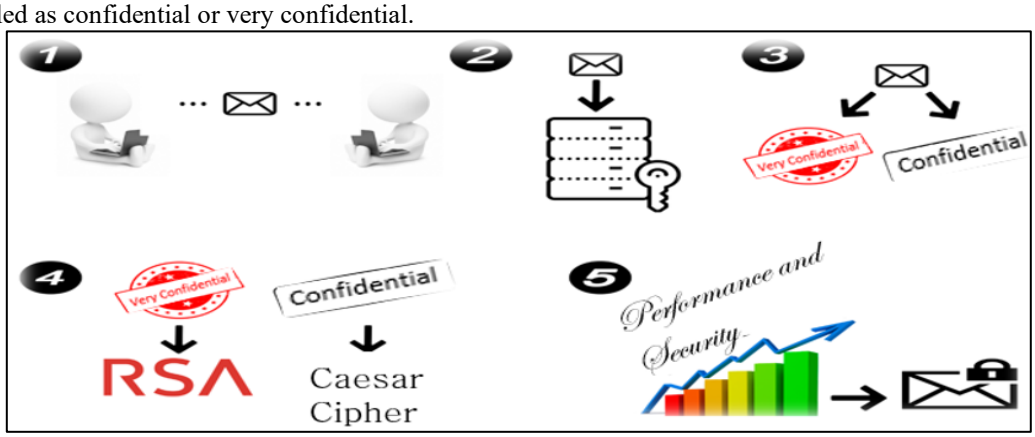

Fig. 7. Resulting Structure.

For this labeling it should be considered the AUC with the highest hit rate, because it is through this classification that the ideal encryption algorithm for the degree of confidentiality determined will be recommended. Thus, avoiding unnecessary consumption of computational resources. In possession of the labeling of the email content, the encryption algorithm is chosen. For emails classified as confidential, it is used the Caesar Cipher method to encrypt its content, and for emails classified as very sensitive, the RSA method is performed. The email is then encrypted by the algorithm considered ideal for labeling, and sent to its intended recipient.

5 Conclusion

It is concluded, based on the experiments carried out, that it is possible to use text pattern recognition algorithms to recommend a custom cryptography, according to what was interpreted, to achieve a better performance in terms of computational resources use.

It is also verified that the proposed structure has the limitation of being a dichotomy, that is, the message classification process is given by two classes only (confidential or very confidential emails). The experiments had a dataset with 57 variables, which corresponds to 57 terms to be considered in the training dictionary. That was a reasonable amount, but in future experiments, with new terms, new collections would be necessary for training and validation. It is relevant to note that the methodology can be used on current and more robust bases, with more samples and monitor words. However, it is pertinent to consider that the computational resource for the experiments, as evaluated in this article, must be consistent with the base to be used.

It was possible to notice the great difference in the execution time between the cryptographic algorithms chosen for the experiments. This is expected because only two methods were used, differentiating them basically in cryptographic techniques (symmetric and asymmetric), where it is expected that the methodology used in the asymmetry technique will require more computational resources as it is a more efficient and robust technique. A recommendation for future works is to use a process of classification of various levels of confidentiality, thus improving the process through intermediate levels of labelling, as well as experiments with dictionary term variation, in this case, variables represented by the terms present in the message.

6 References

Boyd, C. & Mathuria, A., 2003. Protocols Using Shared Key Cryptography. In: *Protocols for Authentication and Key Establishment.* Berlin: Springer, pp. 73-106.

Fawcett, T., 2005. An introduction to ROC analysis. *Pattern Recognition Letters,* Volume 27, pp. 861-874.

Frias-Martinez, E., Sanchez, A. & Velez, J., 2006. Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. *Engineering Applications of Artificial Intelligence,* Volume 19, pp. 693-704.

Gan, M., Wang, C. & Zhu, C., 2016. Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings. *Mechanical Systems and Signal Processing,* Volume 72-73, pp. 92-104.

Jiang, D. et al., 2020. A probability and integrated learning based classification algorithm for high-level human emotion recognition problems. *Measurement.*

Khalifa, O., Islam, M., Khan, S. & Shebani, M., 2004. Communications cryptography. *IEEE,* pp. 220-223.

Li, D., Chen, X. & Huang, K., 2015. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR),* pp. 111-115.

Lima-Junior, F. R. & Carpinetti, L. C. R., 2019. Predicting supply chain performance based on SCOR® metrics and multilayer perceptron neural networks. *International Journal of Production Economics,* pp. 19-38.

Mazurowski, M. A. & Tourassi, G. D., 2009. Evaluating classifiers: Relation between area under the receiver operator characteristic curve and overall accuracy. *2009 International Joint Conference on Neural Networks,* pp. 2045-2049.

Oliveira, C., Xexéo, J. A. M. & Carvalho, C. A. B., 2006. Clustering and categorization applied to cryptanalysis. *Cryptologia,* Volume 30, pp. 266-280.

Pejic-Bach, M., Bertoncel, T., Meško, M. & Krstić, Ž., 2020. Text mining of industry 4.0 job advertisements. *International Journal of Information Management,* pp. 416-431.

Souza, F. H. B. et al., 2019. Risk Prediction For Surgical Site Infection In Craniotomy Patients. *Antimicrobial Resistance & Infection Control,* p. 34.

Souza, W. A. R., De Carvalho, L. A. V. & XEXÉO, J. A. M., 2009. Ciphertexts Clustering is Equivalent to Plaintexts Clustering. *Seventh Brazilian Symposium in Information and Human Language Technology.IEEE.,* pp. 44-52.

Thome, A. C. G., 2012. SVM Classifiers – Concepts and Applications to Character Recognition. In: *Advances in Character Recognition.* London: InTech, pp. 25-50.