Article in Press IJIMAI journal

Deobfuscating *Leetspeak* With Deep Learning to Improve Spam Filtering

Iñaki Vélez de Mendizabal[1], Xabier Vidriales[1], Vitor Basto-Fernandes[2], Enaitz Ezpeleta[1], José R. Méndez[3,4,5,*], Urko Zurutuza[1]

[1] Mondragon Unibersitatea, Faculty of Engineering, Electronics and Computing Department. Loramendi 4, Modragon 20500 Gipuzkoa (Spain)

[2] Instituto Universitário de Lisboa (ISCTE-IUL), University Institute of Lisbon, ISTAR-IUL, Av. das Forças Armadas, 1649-026 Lisboa (Portugal)

[3] Department of Computer Science, ESEI - Escola Superior de Enxeñaría Informática, Universidade de Vigo, Campus Universitario As Lagoas s/n, 32004 Ourense (Spain)

[4] CINBIO - Biomedical Research Centre, Universidade de Vigo, Campus Universitario Lagoas-Marcosende, 36310-Vigo, Pontevedra (Spain)

[5] SING Research Group, Galicia Sur Health Research Institute (IIS Galicia Sur), Hospital Álvaro Cunqueiro Bloque técnico, Estrada de Clara Campoamor, 36312-Vigo, Pontevedra (Spain)

Received 31 March 2022 | Accepted 30 May 2023 | Early Access 7 July 2023

Abstract

The evolution of anti-spam filters has forced spammers to make greater efforts to bypass filters in order to distribute content over networks. The distribution of content encoded in images or the use of Leetspeak are concrete and clear examples of techniques currently used to bypass filters. Despite the importance of dealing with these problems, the number of studies to solve them is quite small, and the reported performance is very limited. This study reviews the work done so far (very rudimentary) for Leetspeak deobfuscation and proposes a new technique based on using neural networks for decoding purposes. In addition, we distribute an image database specifically created for training Leetspeak decoding models. We have also created and made available four different corpora to analyse the performance of Leetspeak decoding schemes. Using these corpora, we have experimentally evaluated our neural network approach for decoding Leetspeak. The results obtained have shown the usefulness of the proposed model for addressing the deobfuscation of Leetspeak character sequences.

Keywords

Convolutional Neural Networks, Deep Learning, *Leetspeak*, Spam Filtering, Text Deobfuscation.

DOI: 10.9781/ijimai.2023.07.003

I. Introduction

Currently, the Internet is one of the most widely used means of communication for exchanging personal (e.g. recreational activities) and corporate information (e.g. business topics). In July 2020, there were more than 4.57 billion Internet users, of which almost 4 billion were using social media services (https://www.statista.com/statistics/617136/digital-population-worldwide/). Internet users can enjoy the speed and simplicity of exchanging information, shopping online or contacting other users. However, some users use the Internet unethically for their benefit, degrading the experience of other users. In particular, one of the most annoying abuses is the distribution of inappropriate and unsolicited content (spam) through communication services based on the exchange of text messages such as classic email [1], [2], social networks [1], [3] or instant messaging [4], [5].

The growth of spam on the Internet has generated the need to develop sophisticated text classification techniques that must be highly reliable and fast to operate. They are used to automatically classify messages into two different spam and ham (legitimate) categories by combining information retrieval [6] (IR) and Machine Learning (ML) [7]. Many text classification approaches have been widely applied to address the problem. Some initial approaches exploited Bag of Words (BoW) representation schemes (using frequency, binary or inverse document frequency values) in conjunction with different types of classifiers, including (i) Naïve Bayes [8], [9], (ii) memory based approaches [10], (iii) decision trees [11], [12], Support Vector Machines (SVM) [13], Artificial Neural Networks (ANN) [14], logistic regression [15], Artificial Immune Systems (AIS) [16], Boosting of trees [17] and other hybrid methods. The latest advances to improve the performance of this type of classifiers consist of the use of synsets obtained from ontological dictionaries such as Wordnet [18] and Babelnet [19] and different types of semantic processing of words [20]–[22].

In the context of the fight against spam, spammers introduced a lot of tricks to avoid spam filters. One of the best known was the use of attached images which became popular in 2007 [23]. This method relates to attaching images that cannot be processed by text classifiers; but are human understandable spam texts. Fig. 1 shows images with embedded texts that are clearly spam and will not be analysed by text filters.

E-mail address: moncho.mendez@uvigo.es

Please cite this article in press as:
I. Vélez de Mendizabal, X. Vidriales, V. Basto-Fernandes, E. Ezpeleta, J. R. Méndez, U. Zurutuza. Deobfuscating *Leetspeak* With Deep Learning to Improve Spam Filtering, International Journal of Interactive Multimedia and Artificial Intelligence, (2023), http://dx.doi.org/10.9781/ijimai.2023.07.003

To combat this type of spam, some researchers took advantage of Optical Character [24] Recognition (OCR) which was initially effective in identifying some words in the image. Later, Battista *et al.* [25] showed how to evade anti-spam filters using text obfuscation techniques in the attached images. To increase the difficulty of identifying the text embedded in the images, spammers add noise to the image [26] (see right image in Fig. 1). More recently, new image-based obfuscation tricks were developed (e.g. as CAPTCHA -Completely Automated Public Turing test to tell Computers and Humans Apart- [27], which can display text and make it unreadable for automatic text recognition systems). However, the latest advances in ANNs have allowed the recognition of the texts [28], [29] included in these CAPTCHAs.

Fig. 1. Examples of images attached to spam messages. These images are part of Image Spam Dataset (https://www.cs.jhu.edu/~mdredze/datasets/image_spam/).

Another important challenge in spam filtering is the recognition of *Leetspeak* (also known as *leet, leet text* or *1337*). This type of slang writing has been used since 1980 [30] and consists of replacing some characters with visually similar symbols so that the reader can understand the message. This type of encoding achieves two simultaneous effects: (*i*) it prevents the classifier from recognising, tokenizing and processing the word and (*ii*) it produces a Bayesian Poisoning [31] attack that inserts random and apparently harmless words into spam messages, causing a spam email to be incorrectly classified as ham. Table I presents twelve *Leetspeak* representations for the word “viagra” (which is often included in spam messages) out of the approximately 600 trillion possible forms for this word. Each column in Table I shows possible replacements for a single character in the word.

TABLE I. Examples of *Leetspeak* Forms for the Word “Viagra”

Original Word	Transformation examples
viagra	\/iagra, |/iagra
viagra	v1agra, v¡agra
viagra	vi4gra, vi/\gra
viagra	via6ra, via(_-ra
viagra	viag12a, viag/2a
viagra	viagr/\, viagr/-\

Table I shows, *Leetspeak* exploits punctuation marks or symbols to hide characters. The replacements made cause misrecognition and misrepresentation of the word during the classification process; therefore, the spammer can bypass spam filters. Using *Leetspeak*, any character (e.g. “A”) can be encoded in many ways and using a different number of symbols (“/-\”, “4”, “|-\”, ...). As *Leetspeak* does not consist of a limited set of symbols, it cannot be solved using a dictionary.

Some previous studies have addressed this problem. Tundis *et al.* designed a convolutional neural network (CNN) to directly classify texts using *Leetspeak* encoding [32]. The use of direct text classification strategies has limitations since the response obtained is not justified. Instead of directly classifying the text, it would be desirable for CNN to allow decoding of the hidden characters in order to provide a solution more understandable from a human point of view. These types of solutions are included in explainable artificial intelligence (XAI) [33]. Subsequently, the same authors proposed a new algorithm for the classification of obfuscated texts that meet the principles of XAI [34]. To do so, they designed a rule-based algorithm in which they exploit a low-precision CNN (*rule-2*) that was created using Chars74K [35][1] image dataset and a collection of images representing non-english characters. Authors tested their CNN using a train-test experiment with their dataset (Chars74K + non-english chars) achieving a performance up to 94,3 percent. However, the performance of their CNN is not measured by classifying *Leetspeak* sequences. In the context of this study, we have trained different CNN models to identify obfuscated characters using the Chars74K dataset for training. All these models returned low *accuracy* scores (in the interval of 42%-52%). The combination of strategies (rules) seems to have allowed the authors to improve the quality of the results obtained. Taking into account the advances achieved in the context of Deep Learning (DL) applied to solving similar problems [36]–[38], we believe that we can obtain performance improvements by creating CNN models from better training data.

In this study, we are introducing a new computer vision approach based exclusively on the use of a CNN model [39], [40] to decode *Leetspeak*. It is able to accurately identify sequences of *Leetspeak* encoded characters represented as images. Using this approach, we are able to recognize the obfuscated words and thus make the full text available to the spam filter. For the implementation, we used TensorFlow [41] and Keras [42]. Our contributions are: (*i*) an image database used for training CNNs for *Leetspeak* deobfuscation, (*ii*) an empirical demonstration that *Leetspeak* recognition can be accurately performed using only CNNs and (*iii*) datasets for the evaluation of *Leetspeak* decoding schemes.

The rest of the manuscript is structured as follows: Sections II and III describe the materials and methods used to complete this study; Section IV presents and analyses the experimental results and finally, Section V describes the conclusions and future research directions.

II. Materials

Currently, there is no dataset available that contains text messages with words obfuscated using *Leetspeak*. In order to create DL models to decode *Leetspeak* character sequences, we had to create a large set of character images to train neural networks (create models) for the task of recognizing the obfuscated characters. The process of creating the image database is described in first subsection. Additionally, we had to generate a new dataset containing obfuscated messages that can be used as a basis for experimentation on this problem. The second subsection describes the process followed to obtain a dataset containing obfuscated messages using *LeetSpeak*.

[1] Available at http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

A. Training Image Database

This paper introduces a computer vision system based on the use of DL to identify obfuscated characters. The process of creating models capable of decoding *Leetspeak* requires the existence of a set of labelled images in which characters are represented. Following the results of the study conducted by Tundis *et al.* [34], we evaluated the Chars74K image dataset. However, this dataset is oriented to assist in the character recognition in natural images and does not fully fit the target of our study. To validate this statement, we trained some models using the Chars74K dataset and applied them for *Leetspeak* deobfuscation achieving classification accuracies in the interval of 42%-52%. Therefore, we have created a database of character images that will be used to train more efficient models.

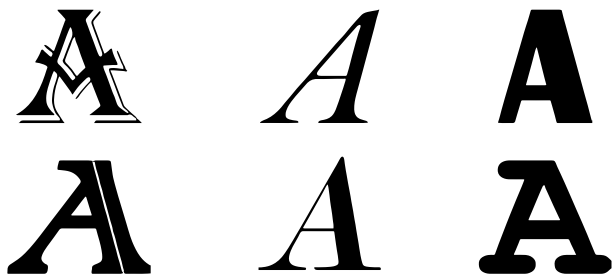

Fig. 2. Examples of images labelled with ‘A’ character.

Our image database was generated by representing each character (‘A’-‘Z’) using 158 different computer fonts and regular, italic, bold and italic+bold styles. The images were obtained at a resolution of 100x100 pixels. Fig. 2 shows some of the images included in the database and labelled with the character “A”.

We improved our image database by adding images extracted from an English handwriting *Dataset*. The resulting image database has a balanced number of images. For each of the 26 characters (‘A’-’Z’) we obtained at least 632 different images and up to 767.

This set of images has been made available in the community section of Mondragon Unibertsitatea website and in Zenodo [43].

B. Datasets for Evaluating Text Deobfuscation Methods

This subsection describes the method designed to obtain corpora in which spam texts may contain obfuscated words and thus be suitable for evaluating the performance of new deobfuscation processes. To do so, we take advantage of well-known and publicly available spam corpora. Table II compiles a set of well-known corpora that provides some interesting features such as content description, ham/spam ratio and the Universal Resource Locator (URL) where the dataset is available.

As shown in Table II, a large collection of datasets with different sizes and contents are available to test the performance. We selected two datasets with classified YouTube comments (YouTube Comments Dataset and YouTube Spam Collection Dataset) that we had used in a previous study [44]. In this study, we only used a subset of 4000 comments from YouTube Comments Dataset (1000 spam and 3000 ham) while the YouTube Spam Collection Dataset was fully used. As the same datasets are used in both studies, it is possible to compare the results obtained. In addition, to extend the study to the email domain, we also selected two medium-sized email datasets (CSDMC 2010 Spam Corpus and TREC 2007 Public Corpus). In this study, the CSDMC 2010 dataset is fully used, while 4327 (32% of them spam) messages were randomly selected from the TREC 2007 dataset. Therefore, both email corpora have the same length and ham/spam ratio.

Once base datasets were selected, we designed an algorithm to create obfuscated contents to be used for the evaluation of our proposal. Table III exemplifies some replacements used in *Leetspeak* to obfuscate the characters of the spam messages.

The obfuscation algorithm implementation involves the following steps: (*i*) randomly select one word to obfuscate in each group of seven words, (*ii*) randomly select a character from the word to be obfuscated (*iii*) applying one of the possible character replacements (see Table III) and (*iv*) repeat the process starting from the last previously selected word until the full content of the message is processed.

TABLE II. Publicly Available Spam Datasets

Dataset	Content description	Spam ratio	URL
British English SMS corpora	875 SMS	48% spam	https://mtaufiqnzz.files.wordpress.com/2010/06/british-english-sms-corpora.doc
Bruce Guenter spam collection	>3,000,000 emails	100% spam	http://untroubled.org/spam/
Clueweb 09	1,040M websites (HTML)	unknown	http://www.lemurproject.org/clueweb09.php/
Clueweb 12	870M websites (HTML)	unknown	http://www.lemurproject.org/clueweb12.php/
Common Crawl Data	9 Billion in 2014 and increasing websites (HTML)	100% spam	http://commoncrawl.org/
CSDMC 2010 Spam Corpus	4327 emails	32% spam	http://csmining.org/index.php/spam-email-datasets-.html
DC 2010 / EU 2010	23M websites (HTML)	unknown	https://dms.sztaki.hu/en/letoltes/ecmlpkdd-2010-discovery-challenge-data-set
Enron email	619,446 emails	0% spam	http://www.cs.cmu.edu/~enron/
HSpam14.s2	14M Twitter messages (tweets)	unknown	https://doi.org/10.1145/2766462.2767701
Ling spam	2,893 emails	16% spam	http://csmining.org/index.php/ling-spam-datasets.html
SpamAssassin	6,047 emails	31% spam	http://spamassassin.apache.org/old/publiccorpus/
Spam Corpus	4,027 emails	34% spam	https://github.com/hexgnu/spam_filter/tree/master/data
SMS Spam Collection v.1	5,574 SMS	13% spam	https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
TREC 2007 Public Corpus	75,419 emails	66% spam	http://plg.uwaterloo.ca/~gvcormac/treccorpus07/
Webspam-uk 2007	105,896,555 websites (HTML)	unknown	http://chato.cl/webspam/datasets/index.php
Websmap-uk 2011	3,766 Web websites (HTML)	53% spam	https://sites.google.com/site/heiderawahsheh/home/web-spam-2011-datasets/uk-2011-web-spam-dataset
Webb spam 2011	330.000 websites (HTML)	unknown	http://www.cc.gatech.edu/projects/doi/WebbSpamCorpus.html
YouTube Comments Dataset	6M Youtube comments	7% spam	http://mlg.ucd.ie/yt/
YouTube Spam Collection Dataset	1,956 Youtube comments	49% spam	https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection

TABLE III. Examples of Character Substitutions in *Leetspeak*

<table>
<tr><td>a</td><td>4, /\, /-\,|-\</td><td>n</td><td>|\|, /\/, [\], , (\), //\\//</td></tr>
<tr><td>b</td><td>8,|3, ß,]3,]8, |8, !3</td><td>o</td><td>0, (), [], ø</td></tr>
<tr><td>c</td><td>(, ¢, <, [, ©, ç, ¢, {</td><td>p</td><td>|°, |?</td></tr>
<tr><td>d</td><td>[), |), |], [>,])</td><td>q</td><td>“(_,)”, “()”</td></tr>
<tr><td>e</td><td>€, 3, [- ,£</td><td>r</td><td>/2, |2, 12</td></tr>
<tr><td>f</td><td>|=, /=, |#, f</td><td>s</td><td>5, $, §, _/¯</td></tr>
<tr><td>g</td><td>6, (_+, , (_-</td><td>t</td><td>+, , ¯|¯, †</td></tr>
<tr><td>h</td><td>#, /-/, [-],]-[,)-(, (-), :-:, |~|, |-|,]~[, }{</td><td>u</td><td>|_|, _/, (_), µ, /_/,]_[</td></tr>
<tr><td>i</td><td>1, !, |, ¦,], :</td><td>v</td><td>\/, |/</td></tr>
<tr><td>j</td><td>¿, , _/, _), 7</td><td>w</td><td>\/\/, \^/, _|_/, _:_/, |/\|, ‘//</td></tr>
<tr><td>k</td><td>|{, |<, |(, }<</td><td>x</td><td>)(, %, ><</td></tr>
<tr><td>l</td><td>|_, []_, [_, 1_</td><td>y</td><td>‘/, ¥</td></tr>
<tr><td>m</td><td>|\/|, /\/\, , (\/), , [\/], //\\//\\, /^^\,</td><td>z</td><td>7_, 2, >_</td></tr>
</table>

When *Leetspeak* is used consciously, it is very likely that all changes applied to a particular character (e.g., “A”) are always the same (e.g., “4”). However, when *Leetspeak* is used to avoid spam filters, some randomly selected characters are automatically replaced by one of its Leetspeak translations. The presented obfuscation method performs the replacements completely random using all possible replacements (see Table III).

The four datasets generated (CSDMC 2010 *Leetspeak*, TREC 2007 Public Corpus *Leetspeak*, YouTube Comments Dataset *Leetspeak*, YouTube Spam Collection Dataset *Leetspeak*) have been shared in a public repository on the website of Mondragon Universitatea (https://mondragon.edu) and Zenodo [45].

III. Methods

Our proposal involves the application of DL strategies for the identification of obfuscated characters. This section explains the identification of *Leetspeak* sequences in text (Subsection A), our proposal to decode them (Subsection B) and the experimental protocol designed for evaluation purposes (Subsection C).

A. Leetspeak Sequence Identification

The deobfuscation problem is identifying the character in the text that best matches a particular sequence of Leetspeak characters. The identification of *Leetspeak* sequences is done by detecting non-alpha characters included in words (sequence of characters that do not contain spaces). In particular, we search for the first and the last non-alpha characters in a word and select the characters included between them as a *Leetspeak* sequence.

TABLE IV. Examples of Obfuscated Leetspeak Character Sequences

<table>
<tr><th>Obfuscated character</th><th>Generated image</th><th>Identified character</th></tr>
<tr><td>|_</td><td>|_</td><td>L</td></tr>
<tr><td>†</td><td>†</td><td>T</td></tr>
<tr><td>€</td><td>€</td><td>E</td></tr>
<tr><td>|</td><td>|</td><td>I</td></tr>
<tr><td>[\]</td><td>[\]</td><td>N</td></tr>
</table>

Once the obfuscated character has been detected and isolated, it is transformed into an image (see examples included in Table IV).

Then, the images are classified using neural networks (DL) for the identification of the obfuscated character. The identified character is used to rewrite the original word and the process starts again until the end of the message. Once a message has been fully decodified, its classification can be successfully performed by taking advantage of common text classification processes. The following subsection explains the DL scheme used to decode *Leetspeak* character sequences.

B. Character Identification Model

For the identification of characters, we take advantage of an image recognition system that does not require the use of static dictionaries. The recognition of each obfuscated character involves looking at some specific sequences of punctuation marks and numbers that are visually similar to the target character. The sequences used to encode a character can be of different length. For example, the ‘V’ character may consist of two consecutive punctuation marks (i.e. a backslash and a slash, ‘\/’). However, in the case of the character ‘H’, it is more common to use three punctuation marks (i.e. ‘]-[‘). Furthermore, our proposal should also recognize new *Leetspeak* variants used to obfuscate words. Keeping in mind these considerations, our proposal includes a CNN. Table V provides detailed information of the layers that make up the CNN design.

TABLE V. Layer Details of Our CNN Used for Character Identification

#	Convolution
1	Conv2D(filters 32, kernel_size (3,3), activation_function=relu, stride=(1,1) MaxPooling2D poolsize=(2,2)
2	Conv2D(filters 64, kernel_size (3,3), activation_function=relu, stride=(1,1) MaxPooling2D poolsize=(2,2)
3	Conv2D(filters 128, kernel_size(3,3), activation_function=relu, stride=(1,1) MaxPooling2D poolsize=(2,2)
Dropout (0,7) Flatten () Dense (neurons 512, activation_function=relu) Dense (neurons 26, activation_function=softmax)	

As shown in Table V, we have defined our CNN as a stack of alternate layers of Convolution, ReLU and MaxPooling. The shape of the input data is a 100x100 pixels image with a colour depth of 1 byte and the output layer comprises 26 neurons and a “softmax” activation function (computes the probability of identifying a specific text character).

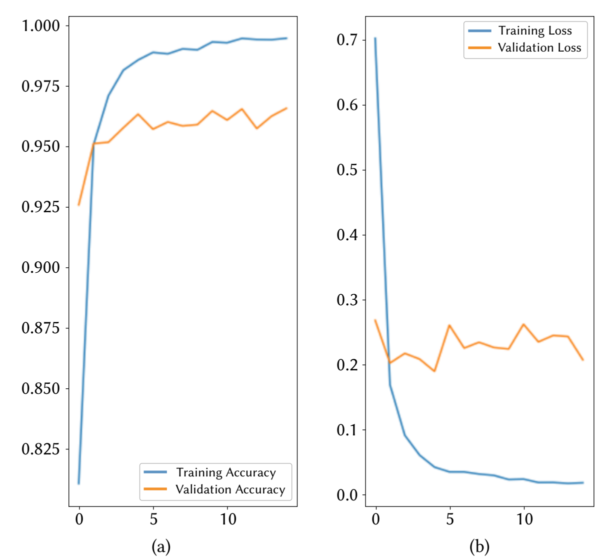

Fig. 3. Evaluation of performance achieved during Training and Validation stage. Parts: a) training and validation *accuracy*, b) training and validation *loss*.

The model has been trained over 15 epochs and 20% of the training dataset (image database described in Subsection II.B) has been reserved to validate model’s performance over the different epochs. In addition, the possibility of adding an early stop as a callback in the training process has been considered to reduce overfitting. However, it was decided to train the model on a certain number of epochs, as the model will try to predict obfuscations formed by characters, but it will only be trained with different variations of real letters. Therefore, in this case it is not essential to apply an early stop to avoid overfitting the model. The *accuracy* and *loss* measurements for training and validation are shown in Fig. 3.

Fig. 3a shows the *accuracy* obtained by the model in each epoch for training and validation datasets. Fig. 3b shows the loss evolution for each epoch. As can be seen, after a few epochs (10) we obtain an *accuracy* close to 90%. After that, the increase in *accuracy* is slower (the neural network needs many epochs to achieve small improvements in *accuracy*).

C. Experimental Protocol

To evaluate the performance of our CNN in a real environment, we created specific test datasets containing spam messages with obfuscated characters (see Subsection II.A). The evaluation was carried out using an experimental protocol (Fig. 4) designed for this purpose.

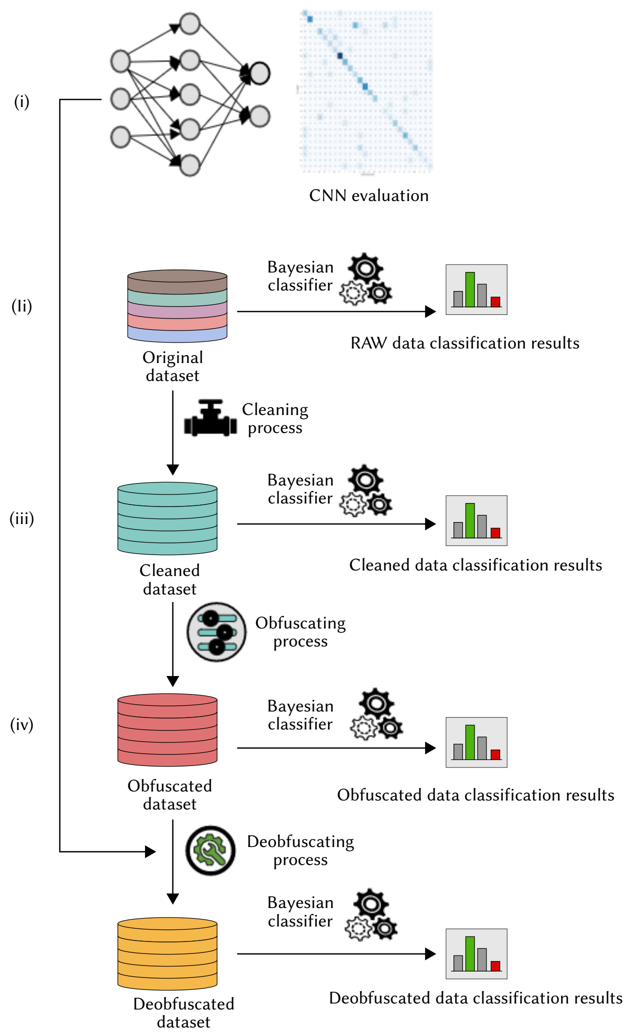

Fig. 4. Experimental protocol.

As shown in Fig. 4, the experiment comprises five steps in which different aspects are evaluated: (*i*) the CNN, (*ii*) the classification of the original datasets (baseline), (*iii*) the classification of the dataset after applying a cleaning process over the original datasets, (*iv*) the classification of the obfuscated datasets and (*v*) the classification of the deobfuscated datasets.

The first step involves training the CNN and evaluating its performance for Leetspeak deobfuscation. For this purpose, we used and analysed a confusion matrix generated by classifying all Leetspeak sequences included in Table III.

During the second step, messages were classified in their original form to obtain a set of baseline performance measures. Due to the large number of different classifiers, we selected the 10 best classifiers identified in the previous work of Ezpeleta *et al.* [44].

The third step consists of identifying and removing non-alphanumeric characters (text cleaning) from the dataset represented in its original form and classifying again the resulting texts. Additionally, in each message, the phone numbers and web URLs included in the message were retained and the rest of the text was converted to lowercase. Pre-processing the messages as described above, we achieved new classification results.

Finally, we classified the obfuscated datasets using the process defined in Subsection II.B (step 4) and the same datasets after being deobfuscated (step 5).

The analysis of results included a comparison of the performance achieved during the last four steps (baseline - step 2, cleaned - step 3, obfuscated - step 4 and deobfuscated - step 5) using standard measures including: *accuracy*, *precision*, *recall* and *f-score* [REF]. We have used a 10-fold cross-validation scheme to run experiments in the last four steps.

IV. Results and Discussion

This section contains the results obtained in the experimentation. First, the implemented CNN was directly evaluated using a confusion matrix. The confusion matrix (Fig. 5) was generated by classifying a dataset of 115 *Leetspeak* sequences.

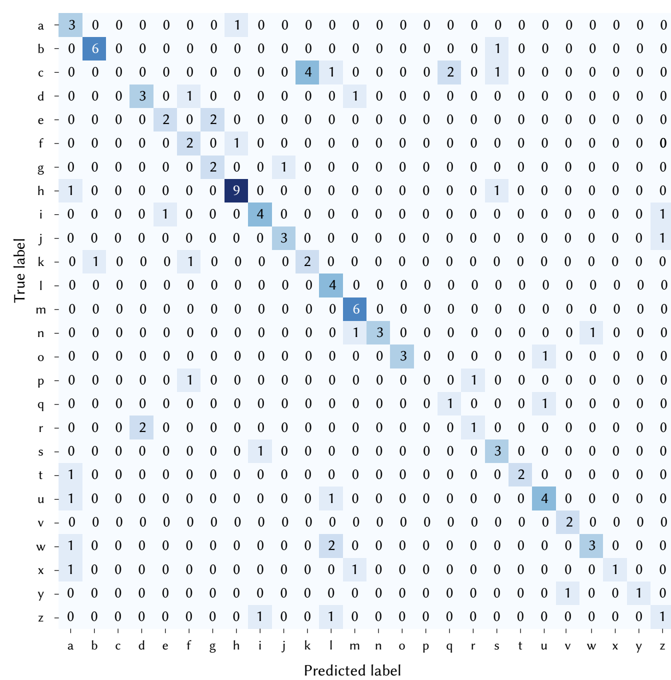

Fig. 5. Confusion matrix achieved using our CNN.

As shown in Fig. 5, although there are some errors, the main diagonal of the confusion matrix shows a large number of hits in recognizing *Leetspeak* sequences. Our CNN has achieved the following performance scores in *Leetspeak* identification: Accuracy = 0.62, Recall = 0.62, Precision = 0.62, f1-score = 0.62, Kappa = 0.61 and Matheus_Coefficient = 0.61.

Then, a practical approach was taken to see how the accuracy of the CNN identification was carried over to the text classification phase. To this end, in fifth step (deobfuscated) the identified *Leetspeak* sequences are replaced by their translations (obtained from the CNN) and then the text is preprocessed and classified. The configurations of classifiers and preprocessing used for the classification process are described in Table VI.

TABLE VI. Legend for Experimental Configurations

Symbol	Meaning
MBM	Multinomial Naïve Bayes
MBMU	Multinomial Naïve Bayes Updateable
CNB	Complement Naïve Bayes
.c	Outupt Word Counts (outwc)
.c	Use a binary representation for tokens (0|1)
.stwv	String to Word Vector
.go	Using the following Weka options (-L -O -W 10000000)
.go	Using default Weka options
.ngtok	NGram Tokenizer 1-3
.ngtok	NGram Tokenizer is not used
.stemmer	Stemmer
.stemmer	Stemmer is not used

Fig. 6 shows the *accuracy* evaluations achieved using the 10 best preprocessing/classification configurations performed in our previous work [44]. The figure has been divided in four separate parts grouping all configurations done by each dataset.

As shown in Fig. 6, the best configurations are those with the original dataset (Baseline and Cleaned). However, when spammers take advantage of *Leetspeak*, using the deobfuscation scheme introduced in this work contributes to improved classification results for all datasets and preprocessing/classification configurations analysed. The use of the deobfuscation schemes allows, in some configurations, to achieve classification results close to those obtained when spammers do not obfuscate the emails (Baseline and Clean). Therefore, the use of CNNs allows good deobfuscation results to be obtained without no need for other complex procedures.

In addition, we also performed an evaluation of the impact of our deobfuscation scheme using precision and recall measures. Table VII shows precision and recall evaluations achieved for all datasets.

As shown in Table VII, the results present the same behaviour as for the *accuracy* evaluation and confirm the utility of the deobfuscation process. Finally, we executed a *f-score* evaluation using all datasets to check whether the deobfuscation was worth according to other criteria. Results are shown in Fig. 7.

As shown in Fig. 7, the *f-score* evaluations through the different scenarios are very similar to previous evaluations obtained for *accuracy*, *recall* and *precision*. The results achieved indicate that substantial performance benefits can be obtained by a deobfuscation process based on the use of CNNs such as the one shown in this study. These successful results are due to the ease with which CNNs automatically detect important features without the need for human supervision. In addition, the use of a wide variety of fonts and styles during CNN training allowed for greater accuracy in the identifying *Leetspeak* sequences.

However, it is very important to select a suitable dataset (such as the one provided as a result of the present research) that allows CNN to learn how to decode *Leetspeak*. Next section shows the main conclusions and outlines future work.

V. Conclusions and Future Work

This study aims to discover mechanisms for automatically decode *Leetspeak* character sequences using only CNN-based models. We provide (*i*) a reliable CNN design for *Leetspeak* deobfuscation processes and its evaluation, (*ii*) an image database that has been used for training the CNN model in this study and (*iii*) four datasets for evaluating *Leetspeak* deobfuscation processes. Through experimental testing, we find that the CNN design and creation processes are able to achieve great performance.

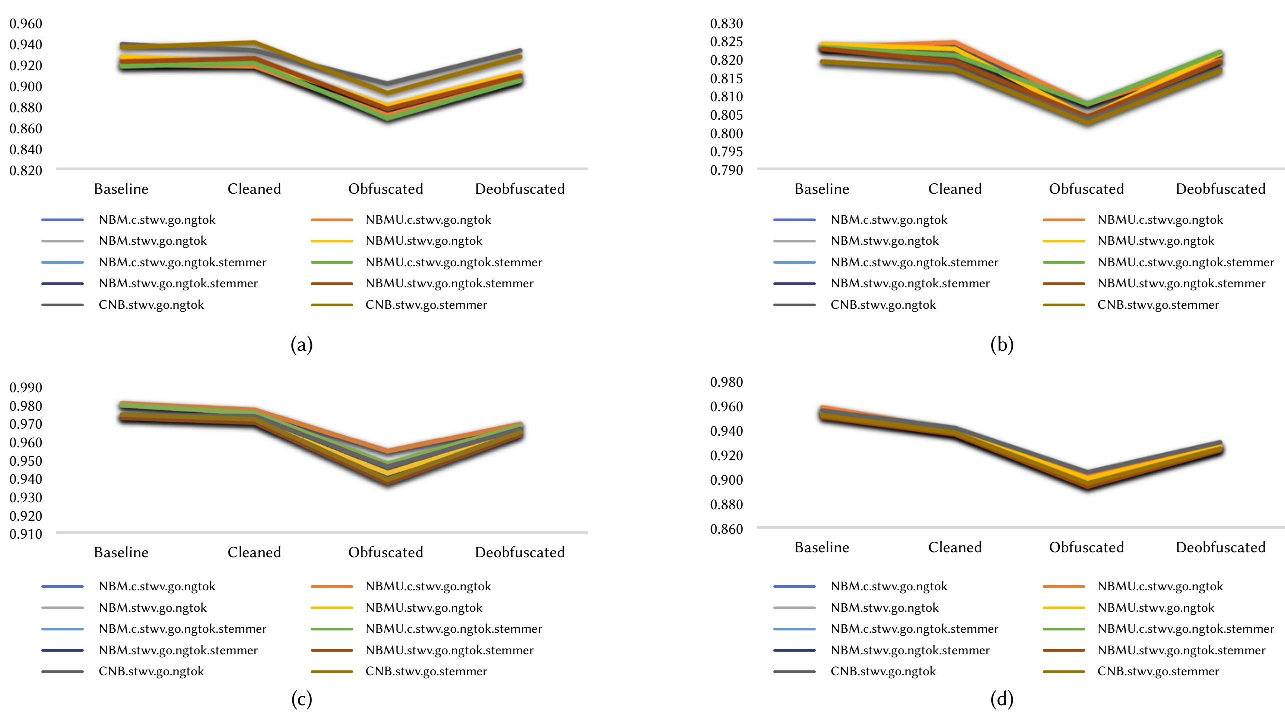

Fig. 6. Experimental results achieved using accuracy measure. Parts: a) Youtube Spam Collection, b) Youtube Comments, c) CMDMC2010 dataset, d) TREC2007 dataset.

TABLE VII. PRECISION AND RECALL EVALUATIONS FOR DATASETS

Classifier/preprocessing configuration	Dataset status	YouTube Comments Dataset		YouTube Spam Collection Dataset		CSDMC 2010		TREC 2007	
	Measure	*precision*	*recall*	*precision*	*recall*	*precision*	*recall*	*precision*	*recall*
NBM.c.stwv.go.ngtok	Baseline	0.801	0.387	0.884	0.972	0.991	0.948	0.987	0.847
	Cleaned	0.798	0.399	0.880	0.974	0.992	0.936	0.991	0.767
	Obfuscated	0.802	0.308	0.807	0.984	0.999	0.861	0.998	0.617
	Deobfuscated	0.808	0.366	0.857	0.979	0.991	0.913	0.993	0.718
NBMU.c.stwv.go.ngtok	Baseline	0.801	0.387	0.884	0.972	0.991	0.948	0.987	0.847
	Cleaned	0.798	0.399	0.880	0.974	0.992	0.936	0.991	0.767
	Obfuscated	0.802	0.308	0.807	0.984	0.999	0.861	0.998	0.617
	Deobfuscated	0.808	0.366	0.857	0.979	0.991	0.913	0.993	0.718
NBM.stwv.go.ngtok	Baseline	0.834	0.371	0.894	0.973	0.992	0.927	0.990	0.829
	Cleaned	0.820	0.373	0.892	0.966	0.994	0.916	0.991	0.763
	Obfuscated	0.809	0.284	0.822	0.982	0.997	0.824	0.997	0.603
	Deobfuscated	0.836	0.357	0.870	0.976	0.994	0.898	0.992	0.714
NBMU.stwv.go.ngtok	Baseline	0.834	0.371	0.894	0.973	0.992	0.927	0.990	0.829
	Cleaned	0.820	0.373	0.892	0.966	0.994	0.916	0.991	0.763
	Obfuscated	0.809	0.284	0.822	0.982	0.997	0.824	0.997	0.603
	Deobfuscated	0.836	0.357	0.870	0.976	0.994	0.898	0.992	0.714
NBM.c.stwv.go.ngtok.stemmer	Baseline	0.822	0.375	0.881	0.973	0.990	0.946	0.989	0.828
	Cleaned	0.805	0.376	0.883	0.978	0.993	0.932	0.993	0.757
	Obfuscated	0.828	0.293	0.805	0.982	0.999	0.839	0.998	0.584
	Deobfuscated	0.826	0.366	0.857	0.978	0.993	0.909	0.996	0.706
NBMU.c.stwv.go.ngtok.stemmer	Baseline	0.822	0.375	0.881	0.973	0.990	0.946	0.989	0.828
	Cleaned	0.805	0.376	0.883	0.978	0.993	0.932	0.993	0.757
	Obfuscated	0.828	0.293	0.805	0.982	0.999	0.839	0.998	0.584
	Deobfuscated	0.826	0.366	0.857	0.978	0.993	0.909	0.996	0.706
NBM.stwv.go.ngtok.stemmer	Baseline	0.847	0.355	0.890	0.969	0.992	0.924	0.991	0.810
	Cleaned	0.820	0.355	0.894	0.971	0.994	0.914	0.991	0.754
	Obfuscated	0.847	0.265	0.817	0.981	0.998	0.806	0.997	0.579
	Deobfuscated	0.856	0.333	0.863	0.978	0.994	0.891	0.994	0.700
NBMU.stwv.go.ngtok.stemmer	Baseline	0.847	0.355	0.890	0.969	0.992	0.924	0.991	0.810
	Cleaned	0.820	0.355	0.894	0.971	0.994	0.914	0.991	0.754
	Obfuscated	0.847	0.265	0.817	0.981	0.998	0.806	0.997	0.579
	Deobfuscated	0.856	0.333	0.863	0.978	0.994	0.891	0.994	0.700
CNB.stwv.go.ngtok	Baseline	0.750	0.415	0.917	0.972	0.991	0.930	0.990	0.833
	Cleaned	0.742	0.412	0.915	0.959	0.994	0.923	0.991	0.777
	Obfuscated	0.734	0.337	0.853	0.976	0.997	0.835	0.997	0.625
	Deobfuscated	0.755	0.398	0.907	0.969	0.993	0.903	0.992	0.728
CNB.stwv.go.ngtok.stemmer	Baseline	0.779	0.388	0.912	0.969	0.992	0.927	0.992	0.818
	Cleaned	0.757	0.396	0.924	0.965	0.995	0.919	0.991	0.758
	Obfuscated	0.757	0.311	0.841	0.975	0.998	0.812	0.997	0.587
	Deobfuscated	0.768	0.384	0.896	0.971	0.994	0.898	0.993	0.707

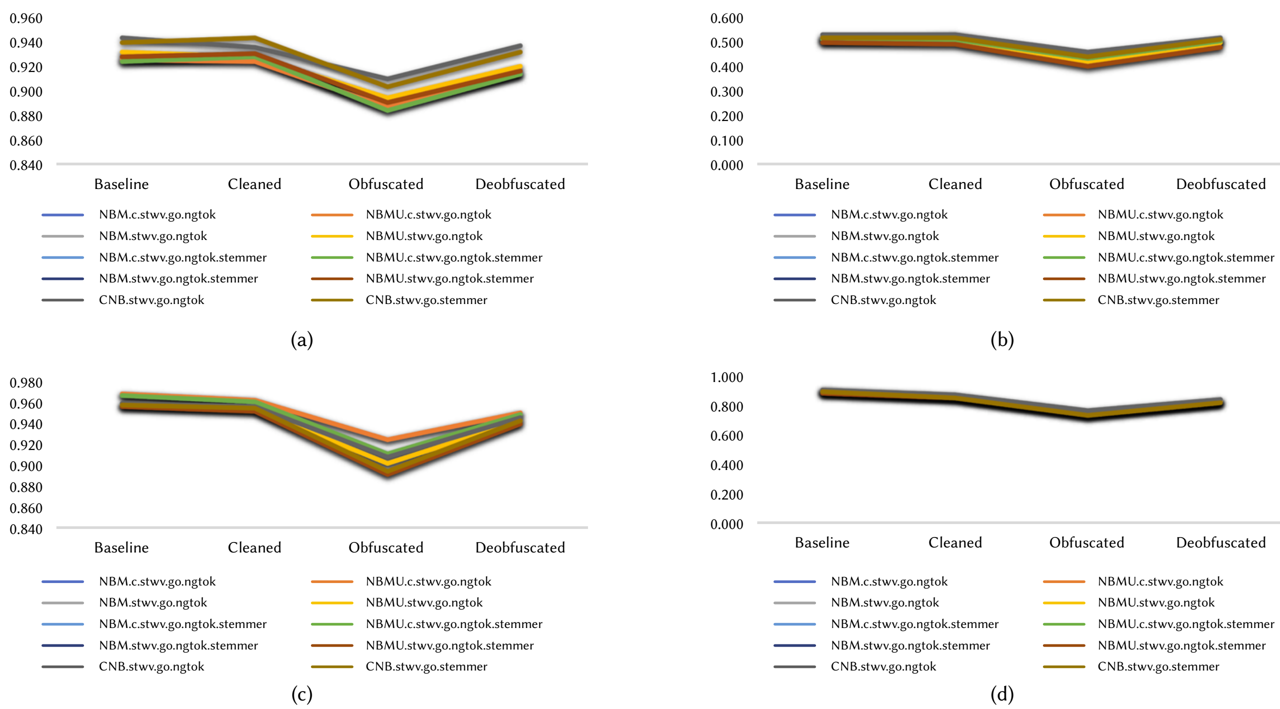

Fig. 7. Experimental results achieved using f-score measure. Parts: a) Youtube Spam Collection, b) Youtube Comments, c) CMDMC2010 dataset, d) TREC2007 dataset.

Analysing the classification rates from the clean text, we can conclude that using *Leetspeak* schemes to obfuscate characters has a huge impact on the performance of all algorithms. By obfuscating characters, spammers are able to completely hide words and make them unusable in spam classification processes. When messages are deobfuscated, the performance of the classifiers increases and reaches, in many cases, the values obtained when messages have not been obfuscated. This fact demonstrates that our proposal can be successfully used to identify the obfuscated characters. However, as shown in Fig. 5, some characters are not correctly identified and further improvements are necessary. Therefore, future work includes extending the image database and improving the CNN architecture to obtain better deobfuscation results.

The main limitation of our proposal is the detection of obfuscated characters containing one single punctuation mark because this requires further analysis. For example, the character H could be obfuscated with a middle hyphen ("-") between two "i" (i.e. "i-i"). This situation could lead to a large number of decoding errors (e.g. "semi-interlaced" being translated into "semhnterlaced", which is incorrect). To address this problem, we consider using dictionary-based schemes (to search whether the word exists with no changes) before using a deobfuscation algorithm. Additionally, we take advantage of the multiple outputs of the CNN (e.g. we consider the five CNN outputs that achieve the highest value) and check the existence of the resulting word in a dictionary. Moreover, the algorithm used to recognize *Leetspeak* sequences also needs to be improved. The one used in this study can only detect one *Leetspeak* sequence per word. Therefore, future work involves improving in several directions (CNN performance, algorithms to detect *Leetspeak* sequences and use of a dictionary) that will lead to significant improvements in the deobfuscation process.

Acknowledgment

Iñaki Velez de Mendizabal, Enaitz Ezpeleta and Urko Zurutuza are part of the Intelligent Systems for Industrial Systems research group of Mondragon Unibertsitatea (IT1676-22), supported by the department of Education, Universities and Research of the Basque Country.

We are supported by the project Semantic Knowledge Integration for Content-Based Spam Filtering, subprojects TIN2017-84658-C2-1-R and TIN2017-84658-C2-2-R, from SMEIC, SRA and ERDF.

Vitor Basto Fernandes acknowledges FCT – Fundação para a Ciência e a Tecnologia, I.P., for its support in the context of project UIDB/04466/2020 and UIDP/04466/2020.

References

[1] M. Chakraborty, S. Pal, R. Pramanik, and C. Ravindranath Chowdary, "Recent developments in social spam detection and combating techniques: A survey," *Information Processing and Management*, vol. 52, no. 6, pp. 1053–1073, 2016, doi: 10.1016/j.ipm.2016.04.009.

[2] S. Suryawanshi, A. Goswami, and P. Patil, "Email Spam Detection: An Empirical Comparative Study of Different ML and Ensemble Classifiers," in *Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing '19*, Tiruchirappalli, India, 2019, pp. 69–74. doi: 10.1109/IACC48062.2019.8971582.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the 1st International Conference on Learning Representations '13*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[4] Y. Cabrera-León, P. García Báez, and C. P. Suárez-Araujo, "Non-email spam and machine learning-based anti-spam filters: Trends and some remarks," in *Proceedings of the Conference on Computer Aided Systems Theory '17*, 2018, vol. 10671, pp. 245–253. doi: 10.1007/978-3-319-74718-7_30.

[5] Z. Liu, W. Lin, N. Li, and D. Lee, "Detecting and filtering instant messaging spam - a global and personalized approach," in *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols '05*, 2005, pp. 19–24. doi: 10.1109/NPSEC.2005.1532048.

[6] C. Manning, P. Raghavan, and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, no. 1, pp. 100–103, 2010.

[7] E. Alpaydin, *Introduction to machine learning*. Cambridge, Massachusetts: MIT press, 2020.

[8] J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," presented at the Second Conference on Email and Anti-Spam

CEAS-2005, California, USA, 2005. [Online]. Available: http://www.ceas.cc/papers-2005/144.pdf

[9] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in *Proceedings of the 3rd Conference on Email and Anti-Spam*, 2006, pp. 28–69. [Online]. Available: http://www.ceas.cc/2006/listabs.html\#15.pdf

[10] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach." arXiv cs.CL/0009009, 2000. [Online]. Available: https://arxiv.org/pdf/cs/0009009.pdf

[11] S. Goyal, R. Chauhan, and S. Parveen, "Spam detection using KNN and decision tree mechanism in social network," in *Proceedings of the 4th International Conference on Parallel, Distributed and Grid Computing '16*, Himachal Pradesh, India, 2016, pp. 522–526.

[12] S. K. Trivedi and P. K. Panigrahi, "Spam classification: a comparative analysis of different boosted decision tree approaches," *Journal of Systems and Information Technology*, vol. 20, no. 3, pp. 298–320, 2018, doi: 10.1108/JSIT-11-2017-0105.

[13] Q. Wang, Y. Guan, and X. Wang, "SVM-Based Spam Filter with Active and Online Learning," in *Proceedings of the 15th Text REtrieval Conference*, Gaithersburg, Maryland, 2006, p. 36. [Online]. Available: https://trec.nist.gov/pubs/trec15/papers/hit.spam.final.final.pdf

[14] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in *Proceedings International Conference on Web Intelligence '03*, Halifax, NS, Canada, 2003, pp. 702–705. doi: 10.1109/WI.2003.1241300.

[15] J. Goodman and W. Yih, "Online Discriminative Spam Filter Training," in *Proceedings of the 3rd Conference on Email and Anti-Spam*, Mountain View, California, 2006, pp. 1–4. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/GoodmanYih-ceas06.pdf

[16] T. Oda and T. White, "Increasing the accuracy of a spam-detecting artificial immune system," in *Proceedings of the 2003 Congress on Evolutionary Computation '03*, Camberra, Australia, 2003, vol. 1, pp. 390–396.

[17] X. Carreras and L. Marquez, "Boosting trees for anti-spam email filtering." arXiv cs/0109015, 2001. [Online]. Available: https://arxiv.org/abs/cs/0109015

[18] C. Fellbaum, "WordNet," in *The Encyclopedia of Applied Linguistics*, C. Chapelle, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012, pp. 1–8. doi: 10.1002/9781405198431.wbeal1285.

[19] R. Navigli and S. P. Ponzetto, "BabelNet: Building a very large multilingual semantic network," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, Uppsala, Sweden, 2010, pp. 216–225.

[20] J. R. Méndez, T. R. Cotos-Yañez, and D. Ruano-Ordás, "A new semantic-based feature selection method for spam filtering," *Applied Soft Computing*, vol. 76, pp. 89–104, 2019, doi: 10.1016/j.asoc.2018.12.008.

[21] E. M. Bahgat and I. F. Moawad, "Semantic-Based Feature Reduction Approach for E-mail Classification," in *Proceedings of the 2nd International Conference on Advanced Intelligent Systems and Informatics '16*, Cairo, Egypt, 2017, pp. 53–63. doi: 10.1007/978-3-319-48308-5_6.

[22] I. Vélez de Mendizabal, V. Basto-Fernandes, E. Ezpeleta, J. R. Méndez, and U. Zurutuza, "SDRS: A new lossless dimensionality reduction for text corpora," *Information Processing & Management*, vol. 57, no. 4, p. 102249, 2020, doi: 10.1016/j.ipm.2020.102249.

[23] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach, "Learning Fast Classifiers for Image Spam," in *Proceedings of the 3rd Conference on Email and Anti-Spam '07*, Mountain View, California, 2007, pp. 1–9. [Online]. Available: https://www.cs.jhu.edu/~mdredze/publications/image_spam_ceas07.pdf

[24] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, "Optical character recognition systems," in *Optical Character Recognition Systems for Different Languages with Soft Computing*, Cham, Switzerland: Springer, 2017, pp. 9–41.

[25] B. Biggio, G. Fumera, I. Pillai, F. Roli, and R. Satta, "Evading SpamAssassin with obfuscated text images," 2007. https://www.virusbulletin.com/virusbulletin/2007/11/evading-spamassassin-obfuscated-text-images/ (accessed Jun. 07, 2023).

[26] J. Evershed and K. Fitch, "Correcting noisy OCR: Context beats confusion," in *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, 2014, pp. 45–51.

[27] E. Bursztein, M. Martin, and J. Mitchell, "Text-based CAPTCHA strengths and weaknesses," in *Proceedings of the 18th ACM conference on Computer and communications security '11*, Chicago, Illinois, USA, 2011, pp. 125–138. doi: 10.1145/2046707.2046724.

[28] J. Wang, J. Qin, X. Xiang, Y. Tan, N. Pan, and College of Computer Science and Information Technology, Central South University of Forestry and Technology, 498 shaoshan S Rd, Changsha, 410004, China, "CAPTCHA recognition based on deep convolutional neural network," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851–5861, 2019, doi: 10.3934/mbe.2019292.

[29] F.-L. Du, J.-X. Li, Z. Yang, P. Chen, B. Wang, and J. Zhang, "CAPTCHA Recognition Based on Faster R-CNN," in *Proceedings of the 13th International Conference on Intelligent Computing Theories and Application '17*, Liverpool, UK, 2017, vol. 10362, pp. 597–605. doi: 10.1007/978-3-319-63312-1_52.

[30] E. Flamand, "Deciphering L33t5p34k Internet Slang on Message Boards," Diss. Ghent University, 2008. [Online]. Available: https://lib.ugent.be/en/catalog/rug01:001414289

[31] J. A. Zdziarski, *Ending spam: Bayesian content filtering and the art of statistical language classification*. San Francisco, California: No starch press, 2005.

[32] A. Tundis, G. Mukherjee, and M. Mühlhäuser, "Mixed-code text analysis for the detection of online hidden propaganda," in *Proceedings of the 15th International Conference on Availability, Reliability and Security '20*, Dublin, Ireland, 2020, pp. 1–7. doi: 10.1145/3407023.3409211.

[33] F. K. Dosilovic, M. Brcic, and N. Hlupic, "Explainable artificial intelligence: A survey," in *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, 2018, pp. 210–215. doi: 10.23919/MIPRO.2018.8400040.

[34] A. Tundis, G. Mukherjee, and M. Mühlhäuser, "An Algorithm for the Detection of Hidden Propaganda in Mixed-Code Text over the Internet," *Applied Sciences*, vol. 11, no. 5, Article ID: 2196, 2021, doi: 10.3390/app11052196.

[35] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proceedings of the 4th International Conference on Computer Vision Theory and Applications '09*, Lisbon, Portugal, 2009, pp. 273–280.

[36] M. Deore and U. Kulkarni, "MDFRCNN: Malware Detection using Faster Region Proposals Convolution Neural Network," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 146–162, 2022, doi: 10.9781/ijimai.2021.09.005.

[37] A. Bhaik, V. Singh, E. Gandotra, and D. Gupta, "Detection of Improperly Worn Face Masks using Deep Learning – A Preventive Measure Against the Spread of COVID-19," *International Journal of Interactive Multimedia and Artificial Intelligence*, pp. 14–25, 2021, doi: 10.9781/ijimai.2021.09.003.

[38] A. Jan and G. M. Khan, "Real World Anomalous Scene Detection and Classification using Multilayer Deep Neural Networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 2, pp. 158–167, 2021, doi: 10.9781/ijimai.2021.10.010.

[39] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, ArticleID 7068349, 2018, doi: 10.1155/2018/7068349.

[40] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep Learning Advances in Computer Vision with 3D Data: A Survey," *ACM Computing Surveys*, vol. 50, no. 2, pp. 1–38, 2018, doi: 10.1145/3042064.

[41] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation '16*, Savannah, GA, USA, 2016, pp. 265–283. Accessed: Mar. 24, 2022. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[42] A. Gulli and S. Pal, *Deep learning with Keras*. Birmingham, UK: Packt Publishing Ltd, 2017.

[43] I. Vélez de Mendizabal, X. Vidriales, V. B. Fernandes, E. Ezpeleta, J. R. Méndez, and U. Zurutuza, "Image dataset to train a deep learning model to decode Leetspeak obfuscated characters." Zenodo, Mar. 21, 2022. doi: 10.5281/ZENODO.6373558.

[44] E. Ezpeleta, M. Iturbe, I. Garitano, I. V. de Mendizabal, and U. Zurutuza, "A mood analysis on youtube comments and a method for improved social spam detection," in *Proceedings of the 13th International Conference*

on Hybrid Artificial Intelligence Systems '18, Oviedo, Spain, 2018, pp. 514–525.

[45] I. Vélez de Mendizabal, X. Vidriales, V. B. Fernandes, E. Ezpeleta, J. R. Méndez, and U. Zurutuza, "Set of obfuscated spam dataset by using LeetSpeak transformations." Zenodo, Mar. 21, 2022. doi: 10.5281/ZENODO.6373653.

Iñaki Vélez de Mendizabal

He is a lecturer and researcher in Mondragon Unibertsitatea. He obtained his degree in Computer Engineering from the University of Mondragon in 2003 and now he is working in his applied engineering PhD. His main research interests are in the areas of computer security and computer networks. At the present his activity focuses on designing developing and implementing semantic analysis systems for improving antispam filters, based on the use of the Multiobjective Evolutionary Algorithms and Deep Learning.

Xabier Vidriales

He was born in the Basque Country (Spain) in 2000. Currently, he is studying Master Degree in Data Analysis, Cybersecurity and Cloud Computing at Mondragon Unibertsitatea and he has been working for three years as a research assistant in the Artificial Intelligence department at the university. He has participated in different projects and worked in several research teams, developing and applying technologies related to spam-filtering, natural language processing and data analysis. His main interests are research on new technologies, especially those related to artificial intelligence, and the development and improvement of data analysis systems.

Vitor Basto-Fernandes

He graduated in information systems in 1995 and got his PhD on multimedia transport protocols in 2006 from University of Minho (Portugal). From 2005 he has been university lecturer, being currently assistant professor with habilitation at ISCTE – University Institute of Lisbon, Portugal. He coordinated a MSc Program in Mobile Computing and was head of the Research Centre in Computer Science and Communications at Polytechnic Institute of Leiria. He participated in several national and international projects in information systems integration, anti-spam filtering and multi-objective optimization, publishes regularly in top-tier journals and conferences, and organized international events in the areas of his research interests, multi-objective optimization, information security and semantic web.

Enaitz Ezpeleta

Dr. Enaitz Ezpeleta is a researcher in the Data Analysis and Cybersecurity Research Group at Mondragon University. He obtained his PhD regarding New approaches for content-based analysis towards Online Social Network spam detection from Mondragon Unibertsitatea in 2016. He is the responsible for coordinating the Artificial Intelligence knowledge area at Mondragon Unibertsitatea and the Master Degree in Data Analysis, Cybersecurity and Cloud Computing. His main research interest applies to spam filtering, Natural Language Processing and data analysis for security. During the last years, Enaitz published over a dozen journal and conference articles and has participated and coordinated work packages and tasks in different public funded research projects, including European, Spanish and Basque Government funded ones.

José R. Méndez

He was born in Galicia (Spain) in 1977. Currently, he works at the computer science department of University of Vigo as associate professor. He worked as a system administrator, software developer, and IT (Information Technology) consultant in civil services and industry during 10 years. He is an active researcher belonging to SING group and, although collaborates in different applications machine learning, his main interests are the development and improvement of anti-spam filters. (http://sing-group.org/).

Urko Zurutuza

Dr. Urko Zurutuza is the principal investigator of the Intelligent Systems for Industrial Systems research group, recognised by the Basque Government Department of Education as a Type A (highest qualification). He coordinates the research activities of this group. He obtained his PhD in January 2008 at Mondragon Unibertsitatea, in a thesis carried out in collaboration with Zürich IBM research Lab (with a pre-doctoral grant from the Basque Government). He has been lecturing in different Telecommunications and Computer Engineering Degrees and Masters, and at the PhD Programme in Applied Engineering at Mondragon Unibertsitatea. He has supervised 8 doctoral theses, and has 3 in progress. He has published more than 20 articles in high impact journals, more than 55 publications in blind peer-reviewed conferences, edited 3 books (2 of them as conference proceedings), and coauthored 7 book chapters. He has experience in 7 European projects funded under programmes such as H2020, CEF Telecom, ECSEL or ARTEMIS. He is also active in knowledge transfer activities, leading more than 40 projects with companies. He has been responsible for International Relations for Telecommunications Engineering, Computer Engineering, and Embedded Systems Master degrees between 2010 and 2018. He is member of the Academic Committee of the Doctoral Programme since 2012. He is member of the Board of Directors of the National Network of Excellence in Cybersecurity Research, and Steering Board member of leading international scientific conferences such as DIMVA or RAID.