Email Spoofing Attack Detection through an End to End Authorship Attribution System

Giacomo Giorgi, Andrea Saracino and Fabio Martinelli
Informatics and Telematics Institute (IIT) of National Research Council, via G. Moruzzi 1, Pisa, Italy

Keywords: Machine Learning, Deep Learning, Privacy-preserving, Email Authorship, Email Spoofing, Spear Phishing.

Abstract: This paper proposes a novel email author verification aimed at tackling email spoofing attacks. The proposed approach exploits an authorship technique based on the analysis of the author's writing style. The problem has been studied under two viewpoints, i.e. the typical sender verification viewpoint, already exploited in previous works, and the sender-receiver interaction verification, which to the best of our knowledge is a novel approach. Hence, we introduced the concept of end-to-end email authorship verification, which is focused on the analysis of the sender-receiver interactions. The proposed method implements a binary classification exploiting both standard machine learning classifiers based on the well-known text stylometric features and deep learning classifiers based on the automatic feature extraction phase. We have used a well-known email dataset, i.e. the Enron dataset to benchmark our approach, with the experiments showing an authorship verification accuracy reaching 99% and 93% respectively for the sender and the end to end verification scenarios. The proposed method has been implemented as an end-user support system in the Android environment for email spoofing attack detection.

1 INTRODUCTION

Email is ubiquitous in our society, and it is an essential part of daily communication, in particular in the workplace where it is still the most common form of communication but also in every online experience where an account is required. As affirmed in (Radicati Group, 2019), in 2019, the total number of business and consumer emails sent and received per day will exceed 293 billion and is forecast to grow over 347 billion by the end of 2023. Despite the benefits provided by email communication, it has also generated new fraud opportunities which can expose the end-user private information to strict security and privacy threats. In recent years the percentage of unsolicited email sent intending to steal private information or harm the recipient device is increasing. Basing on the *Spam and phishing report* published by Kaspersky Lab [1], the average percentage of spam email in global mail traffic in 2018 and Q1 2019 are comprised between 50% and 60%. The most widespread spam attacks are scam emails where the malicious user tries through confidence tricks to deceive the victim into stealing personal information.

One of the forms of scam attack is represented by the *spear phishing*, in which the attacker is intended to steal sensitive information from a specific victim often forging the email header so that the message appears to have originated from someone or somewhere other than the actual source. This type of attack can achieve a high degree of success because people are more inclined to open an email when they think a legitimate source has sent it. The nature of the original Simple Mail Transfer Protocol (SMTP) used in electronic mail transmission (Hoffman, 2002), does not provide an authentication mechanism that can verify information about the origin of email messages. A large number of valid protocols have been proposed to solve the problem such as ESMTP (Myers, 1999), SPF (Wong and Schlitt, 2006), DKIM (Allman et al., 2007), DMARC (Kucherawy and Zwicky, 2015). Nevertheless, the original SMTP is still more widely used (201,). Therefore a system of email authorship verification based on the writing style analysis can be a valid alternative to support end-user to determine, with a certain confidence degree, whether the email sender is who declares to be. In this paper, we focused on a specific email scam (spear phishing) based on email spoofing attack and we implemented a new support end-user system able to detect such at-

[1] https://securelist.com/spam-and-phishing-in-q1-2019/90795/

Giorgi, G., Saracino, A. and Martinelli, F.
Email Spoofing Attack Detection through an End to End Authorship Attribution System.
DOI: 10.5220/0008954600640074
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 64-74
ISBN: 978-989-758-399-5

tack analyzing the email content. In the paper is given the description of the email scam attack and, as contermeasures, two different scenarios based on email authorship verification are presented: (i) a detection on the server side which can exploits the characterization of the overall writing style of a sender, and (ii) a detection on the client side that marginalizes the characterization of a sender only to a specific receiver (*End to End writing style*). We considered solutions based on machine learning systems experimenting both standard machine learning classifiers based on well-known text stylometric features and deep learning classifiers characterized by an automatic features extraction. To reach the best accuracy has been experimented different training approaches, which consider different subset of the dataset used. The best model has been employed in the realization of a secure email client application for Android as instrument to support the end-user in the detection of suspicious emails. The paper is organized as follows. In Section 2 the background concepts related to the spear phishing attack and an introduction of the authorship problem are explained. In Section 3, the proposed authorship approach is discussed, and the details of the framework applied in two possible scenarios, are provided. In Section 4, the feature-based and the deep learning classifiers used and implemented are detailed. Section 5 provides a description of the dataset used and the experiments conducted. In Section 6 the results obtained are presented and discussed. Section 7 describes a panoramic of authorship works analyzed in the literature. In Section 8, the concluding remark and the possible future work are discussed.

2 BACKGROUND

In this section, the background concepts related to the spear phishing attack and the introduction of the authorship problem are given. In particular, is described the basic concept of the attack showing how it is possible forge an email sender through the email spoofing. Besides, the concept of email authorship is introduced, defining two possible learning writing style approaches.

2.1 Spear Phishing

The spear phishing is a form of email scam intended to steal sensitive information from a specific victim. Unlike traditional spam attacks, spear-phishing are personalized to their victims and messages are modified to specifically address that victim. This type of attack can achieves a high degree of success because people are more inclined to open and reply to an email when they think a legitimate or a trustworthy source has sent it. The majority of spear phishing emails use email spoofing as hacking technique to forge the sender address acting on the email header. Due to the structure of the Simple Mail Transfer Protocol (SMTP) used in the electronic mail transmission, email services by default are not capable of identifying and blocking deceptive emails with a forged sender name or email address.

2.2 Authorship

The Authorship attribution process is defined as the problem of determining the likely authorship of a given document. It can be divided into two subproblems: (i) authorship identification and (ii) authorship verification. The goal of the identification is to predict the author of an unknown text within a closed set of candidate authors where, from the classification point of view, can be viewed as a multi-class text classification task. While the goal of the authorship verification is to predict whether a text is written by the declared author and it can be modeled as a binary classification problem in which we attempt to distinguish a single author (target class) from all other authors (not target class). In literature, the problem has been addressed through a study of the linguistic style of a person taking as assumption that each author has distinctive writing habits which can be represented by writing stylistic features. From our perspective, the writing style of a person, can be divided into two different writing style abstraction level: (i) *individual writing style*, which is related to the generic writing style of a person discernible in every context and (ii) *end to end writing style*, related to a user writing style used only with a specific receipts. The concept of *individual writing style* is related to the fact that it is possible to detect distinctive stylistic features that do not change respect to the context, situation, or recipient. Such independence led to consider the individual writing style as a measurable human trait such as a biometric characteristic. Therefore analyzing text/messages sent by an author to a subset of recipients, it is possible to understand the individual writing style of the sender and infers the author of the text/messages sent to new recipients. The concept of *End to End writing style* is based on the fact that a person can assume different writing style depending on the recipient (e.g., colleague, friend, family member), therefore infer the author of a text/message it is possible only analyzing the interaction sender-receiver in order to learn a custom linguistic fingerprinting for each communication.

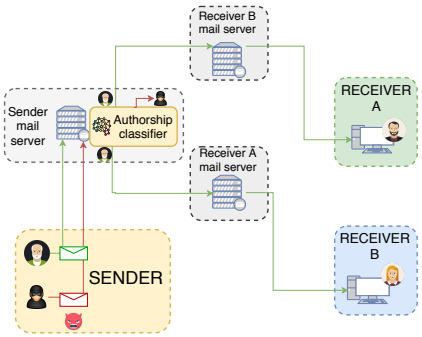

Figure 1: Sender email verification.

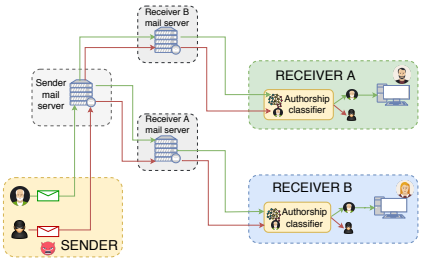

Figure 2: End to end email verification.

3 SCENARIO

Basing on the email system architecture, in this section, two possible scenarios in which the authorship system can be applied, are presented.

3.1 End to End Email Verification

To test the email authorship based on end to end writing style, we consider a context in which, the email authorship system is located on the client-side. In such scenario, only an *end to end communication* between the parties is known and considering the system implemented on the receiver side, only a subset of emails related to the single sender-receiver interaction are known. The system, located on the receiver side, performs a writing style analysis of the received email and assigns it, with a certain confidence degree, a probability of belonging to a legitimate sender. As showed in Figure 2 a malicious user intended to perform a spoofing attack sends an email to the victim declaring to be a legitimate identity, when the email arrives at the receiver side is analyzed to the authorship email classifier which, knowing the end to end writing style of the declared sender, assigns it a low probability to be an email provided by a legitimate identity working as an *Email Anti-Scam* tool.

3.2 Sender Email Verification

If the point of view is moved on the server side, the quantity of information known is not restricted to one single sender-receiver communication but to all the communication which involves the sender. In that case, knowing how the sender writes to all its recipients, the writing style is better characterized, therefore a *individual writing style* can be learned. As showed in Figure 1, a malicious user intended to perform a spoofing attack send an email to the victim declaring to be a legitimate identity, when the email is delivered to the sender email server, the email authorship classifier, knowing the generic writing style of the declared sender analyzes the email and assigns it a low probability to be an email provided by a legitimate identity and send it back.

3.3 Threat Model

In this section is detailed the threat model explaining how the attack can be performed, and the attacker knowledge. The aim of the attacker, which performs a spear-phishing attack, is to steal sensitive information from a specific victim. We assumed that the adversary knows the recipient's email address (victim email address) and the email address of a trusted source for the recipient. In such a case, the attacker can impersonate the trusted source and it can asks sensitive information from that specific victim. we also assumed that the victim and the trusted email accounts are not compromised, whereby the attacker doesn't know the trusted source writing style.

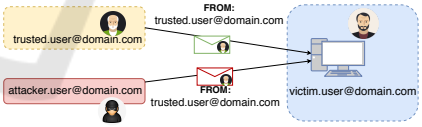

Figure 3: Spear Phishing Attack.

Figure 3, shows a practical example of the spear-phishing. The attacker, knowing only the victim's email address and the email address of a trusted user for the victim, forges the email sender field and sends an email to the victim, impersonating the trusted source.

4 AUTHORSHIP CLASSIFIERS

The email authorship verification can be modeled as a text binary classification problem to distinguish the *target class* (email sent by the declared author) from the *not target class* (email sent by an author different

from who declares to be). The two types of classifiers used in the experiments can be divided into two classes based on the feature extraction method used: (i) features engineering-based, which require domain knowledge of the data to extract features, and (ii) word embedding based, which perform an automatic feature extraction process to learn a words representation from the data.

4.1 Features based Classifier

Features based classifiers used in the experiments, consider a set of linguistic features validated in many authorship verification works (Brocardo et al., 2015), (Zheng et al., 2006). The three main elements that describe a language are lexis, syntax, and semantics. The lexical features are text items that can be a word, part of a word, or a chain of words. Lexical items are the basic building blocks of a language's vocabulary and can be used to measuring the lexical richness of a writing style. By definition, the syntax is the set of rules, principles, and processes that govern the structure of sentences in a given language. Finally, the structural features measure the text organization in terms of the number of sentences or sentence length. The complete list of features used in our classifiers is reported in Table 1. As classifiers, seven different states of art machine learning algorithms are experimented: Nearest Neighbors (Dasarathy, 1991), Radial Basis Function kernel SVM (RBF SVM) (Suykens and Vandewalle, 1999), Decision Tree (Quinlan, 1986), Random Forest (Ho, 1995), AdaBoost (Freund et al., 1999), SGD (Kiefer et al., 1952) and Logistic regression (Peng et al., 2002).

Table 1: Linguistic features.

Category	Feature
Lexical	Number of Characters (C)
	Number of lower Characters/C
	Number of Upper Characters/C
	Number of white space/C
	Number of special Char/C
	Number of Vowels/C
	Frequency of Vowels
	Frequency of non Vowels
	Frequency of special Char
	Number of Words (W)
	Average length per Word
	Number of unique words
	Word(W) - Char (C) ratio
	Most frequently words
	Word 2 and 3-grams
Structural	Number of short words/W
	Number of long words/W
	Number of Sentences (S)
	Average number of words in Sentences
	Number of sentences beginning with Uppercase/S
	Number of sentences beginning with Lowercase/S
Syntactical	Number of punctuation
	Punctuation frequency
	Number of symbols
	Symbols frequency

4.2 Word Embedding Classifier

Neural Networks (NN) require input data as sequences of encoded integers so that each word has to be represented by a unique integer. Therefore it is necessary an encoding schema that represents a sequence of text in an integer vector. Word embedding is a technique for representing words and documents using a dense vector representation (Mikolov et al., 2013), its aim, is a text description where for each word in the vocabulary corresponds a real value vector in a high-dimensional space. The vectors are learned in such a way the words that have similar meanings have similar representations in the vector space. Such text representation is more expressive than more classical methods like bag-of-words, where relationships between words or tokens are ignored, or forced in bigram and trigram approaches. In every network implemented, the embedding layer is initialized with random weights to learn, along with the model, an embedding space for all of the words in the training dataset (custom word embedding). In this way, the vocabulary created reflects the terms contained in the dataset, and it is independent of the language. Two different types of deep learning classifiers based on word embedding have been experimented: (i) Convolutional Neural Network and (ii) Recurrent Convolutional Neural Network.

4.2.1 Convolutional Neural Network

During recent years, Convolutional Neural Network (CNN) has achieved great performances in the Computer Vision field. The extension of the CNN in other fields has proved the effectiveness also in Natural Language Processing (NLP), outperforming state of the art (Zhang et al., 2015). The CNN architecture is composed of a combination of layers that, performing a non-linear operation (convolution and subsampling), can extract essential features from the input data (text sentences in our case). Convolutional layers apply a set of learnable filters to the input with small receptive fields. Such filters are a sort of mask that is applied to the word representation of the input text through a sliding window to detect different text patterns. The set of features extracted through the filters are called *feature map*. The convolutional operation is typically followed by a subsampling operation performed by a max-pooling layer. This layer aims to reduce the dimensionality of the feature map and

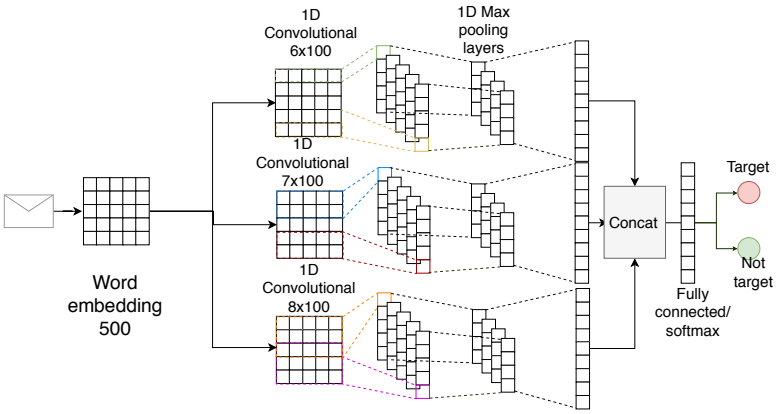

Figure 4: CNN architecture

extract the most significant features. The architecture implemented is composed of three essential part: (i) Custom Word embedding, (ii) Convolutional part, and (iii) Fully connected part. As convolutional neural network, we experimented a multi-channel Convolutional network (Ruder et al., 2016), composed of a custom word embedding of dimension 2000 with 10000 maximal amount of words in the vocabulary, ables to represent each text sequence with maximum length 500 through an integer vector of size 2000. The vector representations are routed to three different Convolutional channels having different learnable filter dimensions (3, 4, and 5) able to extract distinctive feature maps. On the bottom of the network, the feature maps extracted are concatenated, and a fully connected layer with 2 softmax units is applied in order to compute the probability of the input email to belong to the declared sender. The complete Convolutional architecture used in shown in Figure 4.

4.2.2 Recurrent Convolutional Neural Network

Recurrent Neural Networks (RNNs) are successfully applied to sequential information such as speech recognition (Graves et al., 2013), video analysis (Donahue et al., 2015), or time series (Connor et al.,). Different from the traditional neural networks, it considers the dependency between each sequence input value. For this reason, it can successfully be applied to the text analysis context where the text sequences are related to each other. Bidirectional RNNs (Schuster and Paliwal, 1997) is a variant of RNN based on the idea that the output at a specific time is dependent not only on the previous element but also on the future element of the sequence. The network designed and implemented to solve the authorship problem is a combination of a Recurrent (RNN) and a Convolutional (CNN) Neural network (RCNN). The RCNN is able to capture contextual information and text representation, applying respectively recurrent and convolutional layers. The network designed and implemented is composed of four part: (i) Custom word embedding, (ii) Recurrent, (iii) Convolutional and (iv) fully connected part. The text representation through word embedding as in the Convolutional network, is composed of 2000 dimension, a maximum vocabulary size of 10000 and maximum text sequence length set to 500. Figure 5 shows the entire network implemented.

5 EXPERIMENTS AND IMPLEMENTATION

In this section a description of the dataset considering the server and the receiver side are given. In addition are detailed the approaches used to train the classifiers and the implementation of the tool.

5.1 Dataset Analysis

Since emails contain private user information, only a few numbers of datasets that contain personal email labeled with the name of the sender are public. In the following section is described the unique available dataset used to test the authorship email architecture. The *Enron Email Dataset* (Klimt and Yang, 2004) is a collection of emails prepared by the CALO Project (A Cognitive Assistant that Learns and Organizes). It contains data from about 150 users, mostly senior management of Enron company. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. For each of the 150 identity the dataset contains the *inbox folder* and the *sent folder*. The total emails included in the dataset are 517401, sent by 20328 different email accounts to 58564 differ-

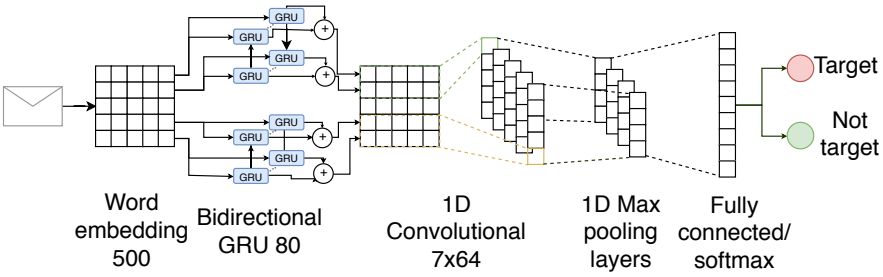

Figure 5: RCNN architecture.

ent receivers. Considering the two scenarios, experimented, have been analyzed the dataset under two viewpoints following explained.

Server Side Dataset. Within the 20328 senders, 136 of them have more than 500 emails sent, and only 67 have more than 1000 emails. Analyzing the email lengths of the dataset, we can identify three different email set: (i) Short emails: emails having less than 20 words, (ii) Medium emails: emails having more than 19 and less than 51 words and (iii) Long emails: emails having more than 50 words.

Table 2: Enron Senders and communications.

Email length	Senders	Sender-Receiver Communications
No constraint	67	256
words >50	49	126
20 <words <50	13	256
words <20	5	256

That analysis shows as the majority of the identity sent long emails followed by the medium emails and only few identities sent short emails. The number of senders having more than 1000 emails considering different length is summarized in the second column of the Table 2.

Receiver Side Dataset. In the receiver scenario, we are interested in considering users that have a considerable number of emails received from the same identity to learn with more accuracy the end to end writing style of the sender toward the receiver. Considering 100 as the minimum number of emails that a single class has to contain to train a classifier, the number of receivers with more than 100 emails received from a single user and more than 100 emails received from other users is 26, while 256 are the total amount of sender-receiver interactions. As showed in the third column of Table 2, from the dataset, considering only communication with more than 100 emails, it is possible extract 256 overall sender-receiver, 126 having long, 256 medium and 256 short length.

5.2 Training and Evaluation

In this section, the training approaches used in both the scenario are detailed.

Sender Email Verification Training. On the server-side, we considered the authorship system on the sender email server, in this way it is make possible to test the learning of the individual writing style of the target sender. For every sender identity, a binary classifier has been trained selecting its inbox emails as a positive class and a list of emails randomly selected from other senders as a negative class. In that scenario, the amount of sender-receiver communications known on the server allow to learn the individual writing style of the sender. During the training, we considered identities having more than 1000 emails sent and for each of one have been trained a binary classifier considering a balanced training set selecting randomly 1000 emails sent by the target class (sender) and 1000 emails randomly selected from the sent emails of other identities of the dataset. As a testing phase, a 10 cross-fold validation has been performed using 100 testing emails for the positive class and 100 emails for the negative class.

End to End Email Verification Training. In the end to end email verification context, as explained in Section 3, the authorship verification system, is located on the receiver side, simulating in such a way the end to end authorship verification. For each recipient identity have been selected a set of sender identities, and in turn, choosing a single target sender (target communication), has been trained a binary classifier using the target emails as positive class and the remaining sender emails as negative class. During the training phase, 256 sender-receiver communications having more than 100 emails, have been considered. A random sub-sampling of the majority class to balance the training set has been performed.

Training Approaches. Two different training approaches, in both the experiments have been used. As shown in Section 5.1, the dataset can be splitted considering different email length. Therefore, as well as the standard training approach, that consider the training data selection independent from the mail length, have been considered a training approach customized for the following subsets: (i) short emails (less than 20 words) (ii) medium emails (between 20 and 50 words) (iii) long emails (greater than 50 words). Each networks' training has been performed on balanced data (number of positive emails equal to the number of negative emails), performing a random subsampling of the majority class when required. A 10 cross fold-validation has been applied during the training phase to have a better evaluation of the machine learning models. The classifiers are evaluated through the computation of the accuracy on the predictions.

5.3 Implementation

The aim of the proposed work is not to build an email authentication system, but we focused on building an alternative instrument to support the end-user in the detection of a possible email spoofing attack. To this end a secure email client application for Android has been developed[2], it works as a standard client email system offering the possibility to connect to the own mail server, download the emails and analyze them with the end to end authorship attribution system.

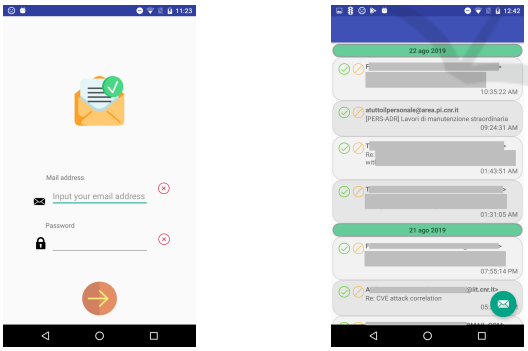

Figure 6: Android secure email client application.

The end-user selecting the list of senders to monitor, launches a training phase on each end to end communication using, if it exists, the past emails exchanged by the parties. When a new email arrives, the system, reading the "declared sender", routed the email to the proper classifier where an analysis of the end to end writing style is performed and it assigns to the email a score that indicates whether the email sender is who declares to be. The classifiers are continuously trained to allow a learning of the end to end writing style over time. The interface of the Android application is showed in Figure 6.

[2]http://github.com/iitcybersecurity/EmailClientSpoofingDetection

6 RESULTS

In this section, the results obtained from the experiments described in Section 5 are shown. In particular, are reported the results obtained in the two scenarios introduced using the proposed training approaches. In addition a discussion of the results obtained comparing the end to end and the sender email verification results, the classification approaches adopted, and the impact of the email length on the accuracy, is given.

6.1 Sender Verification Results

For the server-side scenario, we reported the evaluation of the classifier in terms of accuracy both for the *training independent from the email length* and for the *training based on the email length*. Table 3 shows the accuracy comparison between the classification mechanisms adopted for the training independent from the email length. It shows the overall accuracy and the specific accuracy of each testing set (short, medium, long).

Table 3: Length independent sender results.

Classifier	Accuracy Short	Accuracy Medium	Accuracy Long
RCNN	89%	94%	94%
CNN	90%	95%	95%
Logistic Reg.	92%	95%	96%
Nearest Neigh.	73%	65%	66%
SVM	92%	95%	96%
Decision Tree	77%	87%	93%
Random For.	90%	94%	96%
AdaBoost	83%	92%	95%
SGD	88%	94%	94%

The reported results are measured through the mean accuracy of 67 target senders having more than 1000 emails sent. The results show the low accuracy of each classifier in recognizing the sender identities through short emails. Conversely, higher accuracy for the medium and long test set, has been obtained. Such results can be because the email length influences accuracy until a certain threshold.

Splitting the training set basing on the email length and building a custom classifier for each subset as described in Section 5, we obtained the results reported in Table 4. It shows for every classifier, the average accuracy obtained in recognizing 5, 13 and 49 senders, respectively for the short, medium

and long test set. As in the previous experiment, the lower accuracy is given by the short email set, which does not take advantage of the custom training. Better results in the medium and long email training set, have been reached, where the accuracy increases of 1-2% respect to the training independent from the email length. The results obtained shown as the email length is an important feature to recognize the author of an email and we can deduce that a short email containing less than 20 words, does not include sufficient information for the author verification. Excluding the short email set from the results, it is possible compare the two training approaches tested.

Table 4: Length dependent sender results.

Classifier	Accuracy Short	Accuracy Medium	Accuracy Long
RCNN	89%	96%	95%
CNN	90%	97%	96%
Logistic Reg.	87%	96%	96%
Nearest Neig.	60%	87%	88%
SVM	90%	96%	96%
Decision Tree	79%	90%	93%
Random For.	85%	95%	96%
AdaBoost	79%	94%	95%
SGD	86%	94%	95%

Table 5 shows the comparison between the two training approaches both for the total testing set (short, medium and long) and for the medium and long testing sets. In both cases, performing the email length dependent training method, the word embedding classifiers have an accuracy increment, in fact, considering the CNN classifier, its accuracy goes from 95% to 96.5% in the medium and long test set, while from 93.3% to 94.3% in the total testing set.

Table 5: Sender verification results comparison.

Classifier	Lenght Independent		Length Dependent	
	AVG Med/Long	AVG Short/Med/Long	AVG Med/Long	AVG Short/Med/Long
RCNN	94%	92.3%	95.5%	93.3%
CNN	**95%**	**93.3%**	**96.5%**	**94.3%**
Logistic Reg.	95.5%	94.3%	96%	93%
Nearest Neigh.	65.5%	68%	87.5%	78.3%
SVM	95.5%	94.3%	96%	94%
Decision Tree	90%	85.6%	91.5%	87.3%
Random For.	95%	93.3%	95.5%	92%
AdaBoost	93.5%	90%	94%	89.3%
SGD	94%	92%	94.5%	91.6%

6.2 End to End Verification Results

As in the sender verification scenario, we reported the results for both the training approaches used. Table 6 shows the mean accuracy of each machine learning models computed from the evaluation of every single end to end classifier trained on the sender-receiver communication independently from the email length. The table, as well as, showing the total average accuracy obtained training the overall sender-receiver communications, shows the average accuracy obtained in every subset of the testing set (short, medium and long). From the analysis of the results, it is possible to affirm that the models based on word embeddings outperform the feature engineering based models. Considering the total accuracy, CNN and RCNN provide higher accuracy respect to the features engineering based models achieving as best result 95.3% of accuracy against the 94.2% reached by the Logistic Regression classifier. Analyzing the accuracy computed for each subset, the short email set shows low accuracy in every model. As in the sender verification scenario, the accuracy increase by increasing the email length until a certain threshold and the better accuracy is achieved with the email having length comprised between 20 and 50 words. It is possible to associate the accuracy trend obtained to the fact that short emails do not contain personal writing style features needed to the classifier to discriminate from one communication to another.

Table 6: End to End verification results length independent.

Classifier	Total Accuracy	Accuracy Short	Accuracy Medium	Accuracy Long
RCNN	**95.3%**	**91.2%**	**96.3%**	**97.1%**
CNN	94.8%	92.6%	97.2%	97.4%
Logistic Reg.	94.2%	84.3%	96.5%	96.3%
Nearest Neig.	81.4%	79.1%	85.4%	83.1%
SVM	94.2%	74.8%	98.0%	95.6%
Decision Tree	92.1%	76.3%	93.1%	93.9%
Random For.	93.6%	77.1%	94.6%	95.6%
AdaBoost	92.7%	80.2%	96.7%	94.3%
SGD	94.5%	80.4%	95.4%	96.0%

Table 7, shows the results obtained performing the email length dependent training in each subset defined. As in the sender verification test, the accuracy obtained is higher respect to the training independent approach and it confirms the validity of the training method proposed.

Table 7: End to End verification results length dependent.

Classifier	Accuracy Short	Accuracy Medium	Accuracy Long
RCNN	**91.3%**	**99.2%**	**98.8%**
CNN	92.5%	98.9%	98.6%
Logistic Reg.	85.3%	97.2%	97.7%
Nearest Neigh.	79.4%	86.5%	84.5%
SVM	75.5%	98.1%	97.6%
Decision Tree	77.4%	95.7%	94.6%
Random For.	78.5%	96.2%	97.4%
AdaBoost	80.9%	97.4%	96.8%
SGD	81.3%	98.0%	97.1%

The accuracy increment is assessable discarding the short test set. Taking in consideration the best model (RCNN), it achieves 99.2% and 98.8% of accuracy respectively in the medium and long set, that are better accuracy comparing to the 96.3% and 97.1% reached

with the length independent training.

6.3 Verification Comparison

A comparison between the two authorship approaches, it is possible only testing the classifiers on the same testing set. Considering an end to end communication composed by one "declared sender" and one receiver, it is possible to apply both the sender authorship verification systems respectively trained on the "declared sender", and the specific end to end communication. Therefore, we performed a sender prediction of each end to end testing set using the proper trained sender authorship classifier. The average accuracy on the overall 256 end to end communications using both the strategies is showed in Table 8. The accuracy of the sender classifiers applied to the end to end testing set, is lower in every testing subset respect to the end to end email verification approach. Such behavior is due to the fact that the sender classifier is able to learn an high abstraction level of the identity writing style that is useful to distinguish two different senders which interact with different receivers, but as highlighted by the accuracy differences, such learned degree is not sufficient to distinguish different senders which interact with the same receiver.

Table 8: End to End - Sender verification results comparison.

Classifier	End To End			Sender		
	Short	Medium	Long	Short	Medium	Long
RCNN	**91,3%**	**99,2%**	**98,8%**	**85.4%**	**93.1%**	**92.2%**
CNN	92,5%	98,9%	98,6%	83.6%	93.5%	92.4%
Logistic Reg.	85,3%	97,2%	97,7%	80.5%	87.9%	87.2%
Nearest Neigh.	75,5%	86,5%	84,5%	73.6%	70.3%	71.8%
SVM	77,4%	98,1%	97,6%	74.9%	85.7%	82.6%
Decision Tree	78,5%	95,7%	94,6%	72.6%	86.9%	85.1%
Random For.	78,5%	96,2%	97,4%	75.4%	85.8%	85.7%
AdaBoost	80,9%	97,4%	96,8%	77.9%	88.3%	87.7%
SGD	81,3%	98%	97%	72.9%	87.1%	86.2%

Table 9: Authorship works comparison.

Ref.	Dataset	Text size	Identities	End 2 End Verification	Sender Verification
[2]	Enron	500 chars	87	-	EER 14.35%
[14]	Enron	<95 words	52	-	Accuracy 97%
[22]	Twitter	1000 chars	50	-	Accuracy 76%
Our	Enron	>20 words	67	99%	96.5%

7 RELATED WORK

In this section, the authorship works are presented taking into consideration the differentiation between feature engineering based and deep learning authorship classifiers, as well as the differentiation between authorship for identification and verification. Authorship analysis is a topic widely treated in literature and in particular in forensic linguistics field where the aim is to identify linguistic features that can give information about the identity of an anonymous text. Many works have been done regarding authorship identification, verification, and writing style characterization. The first works on authorship were related to the attribution of an author to a specific textbook or general text document well structured and having a long dimension. The new investigations are focused on authorship analysis of online documents that have reduced text size and in general, not well structured like social messages or emails (Brocardo et al., 2013). The main approach used to solve that problem is to use the stylometric features manually extracted to specify the writing style of a person through traditional machine learning algorithms. The effectiveness of deep learning neural network in Natural Language Processing (NLP), have provided advantages in feature extraction, and some techniques have also been applied in the authorship field. Most of the authorship works are focused on the identification problem (attribution of identity to a given text), in (Zheng et al., 2006) the authors present an online message authorship identification framework based on four types of writing style (lexical, syntactic, structural and content-specific). They experimented with three features based on classification techniques on English online text with an average length of 169 words. They achieved 97% of accuracy in identifying 20 identities through 30, 40 messages per author. In (Shrestha et al., 2017) is presented another work on authorship identification of short messages based on a deep learning model. The authors presented a Convolutional Neural Network for the author attribution of tweets achieving 76% of accuracy for 50 authors with 1000 tweets each. Another authorship subfield studied in short message analysis is the verification problem (verify whether the written text belongs to who declares to be). In such context deep learning models have also been applied to the authorship verification problem for short messages, in (Litvak, 2018) is presented a deep learning model for automatic feature extraction directly from the input text. They implemented a Convolutional Neural Network ables to analyze the raw input email text and extract the discriminate features to verify the genuineness of the author. Table 9 summarizes the comparison between our work and the studies in this field.

8 CONCLUSION

We faced the problem of spear-phishing attack based on the forgery of the sender field contained in the

email. As a countermeasure, we proposed an end-user email support system based on the analysis of the writing style of a person. We presented two possible approaches to solve the problem (i) sender email verification which we exploited the characterization of the overall writing style of a sender and (ii) end to end email verification, which considers the end to end writing style in the sender-receiver communication. As a verification system, we proposed an authorship email verification based on a binary text classifier. We compared two text classification approaches (i) features engineering based and (ii) word embedding based. In both the scenarios experimented are tested two training techniques based on different splitting of the dataset: (i) independent from the email length and (ii) dependent from the email length. The analysis of the results shows: (i) the higher accuracy of the word embedding based classifiers respect to the features engineering based in both the scenarios; (ii) the effectiveness of the training technique based on the dataset splitting dependent from the email length and (iii) the better accuracy obtained by the end to end email verification respect to the traditional sender verification. With the high accuracy reached in the email author verification, it has been proved that the authorship mechanism is a promising support approach to use in contrast to the spear-phishing scam emails.

ACKNOWLEDGEMENTS

This work has been partially supported by H2020 EU-funded projects SPARTA, GA 830892, C3ISP, GA 700294 and EIT-Digital Project HII, PRIN Governing Adaptive.

REFERENCES

Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and Thomas, M. (2007). Domainkeys identified mail (dkim) signatures. Technical report, RFC 4871, May.

Brocardo, M. L., Traore, I., Saad, S., and Woungang, I. (2013). Authorship verification for short messages using stylometry. In *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. IEEE.

Brocardo, M. L., Traore, I., and Woungang, I. (2015). Authorship verification of e-mail and tweet messages applied for continuous authentication. *Journal of Computer and System Sciences*, 81(8):1429–1440.

Connor, J. T., Martin, R. D., and Atlas, L. E. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5.

Dasarathy, B. V. (1991). Nearest neighbor (nn) norms: Nn pattern classification techniques. *IEEE Computer Society Tutorial*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Hoffman, P. (2002). Smtp service extension for secure smtp over transport layer security.

Kiefer, J., Wolfowitz, J., et al. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466.

Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer.

Kucherawy, M. and Zwicky, E. (2015). Domain-based message authentication, reporting, and conformance (dmarc).

Litvak, M. (2018). Deep dive into authorship verification of email messages with convolutional neural network. In *Annual International Symposium on Information Management and Big Data*, pages 129–136. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Myers, J. G. (1999). Smtp service extension for authentication.

Peng, C.-Y. J., Lee, K. L., and Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1):3–14.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Radicati Group, I. (2019). Emailstatistics report, 2019-2023.

Ruder, S., Ghaffari, P., and Breslin, J. G. (2016). Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv preprint arXiv:1609.06686*.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Shrestha, P., Sierra, S., Gonzalez, F., Montes, M., Rosso, P., and Solorio, T. (2017). Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Wong, M. and Schlitt, W. (2006). Sender policy framework (spf) for authorizing use of domains in e-mail, version 1. Technical report, RFC 4408, april.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zheng, R., Li, J., Chen, H., and Huang, Z. (2006). A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American society for information science and technology*, 57(3):378–393.