Technical Disclosure Commons

Defensive Publications Series

January 02, 2019

Visual match of emails or landing pages to detect phishing

Kuntal Sengupta

Vijay Eranti

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Sengupta, Kuntal and Eranti, Vijay, "Visual match of emails or landing pages to detect phishing", Technical Disclosure Commons, (January 02, 2019)
https://www.tdcommons.org/dpubs_series/1836

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Visual match of emails or landing pages to detect phishing

ABSTRACT

In a phishing attack, a perpetrator attempts to obtain the online credentials of a user by impersonating a trusted entity such as a bank, email service provider, etc. Sophisticated phishers attempt to deceive spam filters by structuring the visual look-and-feel of their fake emails to be nearly but not precisely identical to emails sent by a trusted entity, such that the spam filter allows the fake email to reach a user's inbox.

This disclosure applies machine-learning based techniques to assess the visual similarity of genuine and phished emails (or landing pages) for a given brand. The techniques detect visual near-duplicates of a trusted entity's email and thereby achieve resilience against adversarial attacks. The need for use of hand-crafted features to achieve visual-similarity match is eliminated, enabling accurate detection of new genres of phishing email as they surface.

KEYWORDS

- phishing
- spoofing
- deduplication
- deep learning

BACKGROUND

In a phishing attack, a perpetrator attempts to obtain the online credentials of a user by impersonating a trusted entity such as a bank, email service provider, etc. Phishers send emails to unsuspecting users (e.g., bank account-holders, email account holders, etc.) that look nearly identical to those sent by the trusted entity. Email service providers deploy spam/phishing filters to thwart such attacks, e.g., by identifying such emails and classifying them as spam/suspicious.

Currently, such filters use features computed from the text, embedded links, sender domain, visual information (e.g., overall look-and-feel), etc. of an email to decide if the email is a phishing attempt. For example, an email that is visually indistinguishable from one sent by a trusted entity, but with embedded links that point to other online entities rather than the trusted entity, is an indicator of a phishing attempt.

Sophisticated phishers try to deceive spam filters by using a visual look-and-feel to the email that is nearly but not precisely identical to one sent by a trusted entity, such that the spam filter allows the fake email to reach a user's electronic mailbox. Due to the near-identicalness of the email from a phisher to that from a trusted entity, a human reader of such a fake email may be fooled into mistaking it as an email from the trusted entity.

DESCRIPTION

This disclosure applies machine-learning techniques, e.g., deep neural networks, to detect visual near-duplicates or minor variations from a trusted entity's email (or landing site). This problem, also known as brand phishing, is treated as a clustering problem in a manifold, where phishing emails of a particular brand are clustered around the brand's template image. The use of hand-crafted features to achieve visual-similarity match is avoided. For example, as new genres of phished email surface, the machine-learning model is re-trained with the new examples, rather than manually selecting features from the new examples and adding to classifier rules. The described techniques achieve robust hardness against adversarial phishing attacks. The techniques are able to perform the visual classification in a few, e.g., less than ten, milliseconds, and easily scale up even for email service providers that serve billions of emails per day.

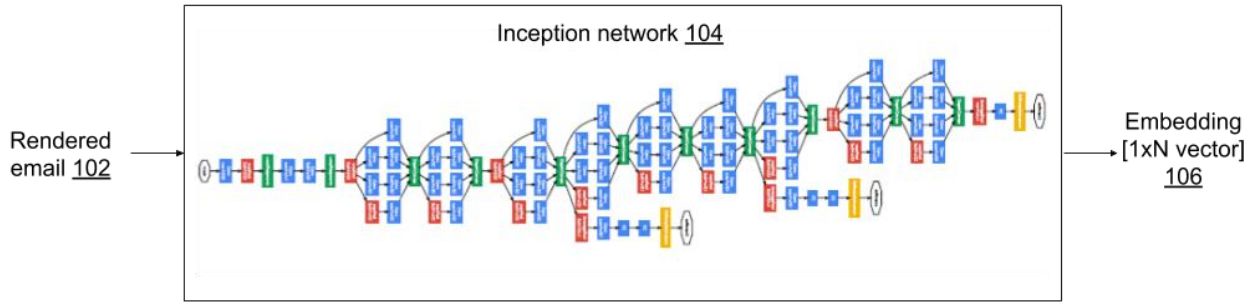


Fig. 1: Use of an inception network to transform a visually rendered email to an embedding

Observing that the problem of computing visual similarity between emails is similar to a face recognition problem, a robust face-recognition machine-learner, e.g., an inception deep neural network, a facenet, a deep convolutional network, etc., is trained on emails. This is shown in Fig. 1, wherein the inception network (104) acts on visually rendered email (102) to produce as output an embedding (106), which is typically a vector quantity. If a technique similar to facenet is used to detect visual similarity between a branded email from a trusted entity and a phishing email then the problems of: similarity - is this the same person (email)?, recognition - who is this person (email)?, and clustering - find common people among these faces (emails), are unified. The face-recognition neural network, adapted to emails, is based on learning a Euclidean embedding per image. The network is trained such that the squared L2 distances in the embedding space correspond directly to visual similarity of emails.

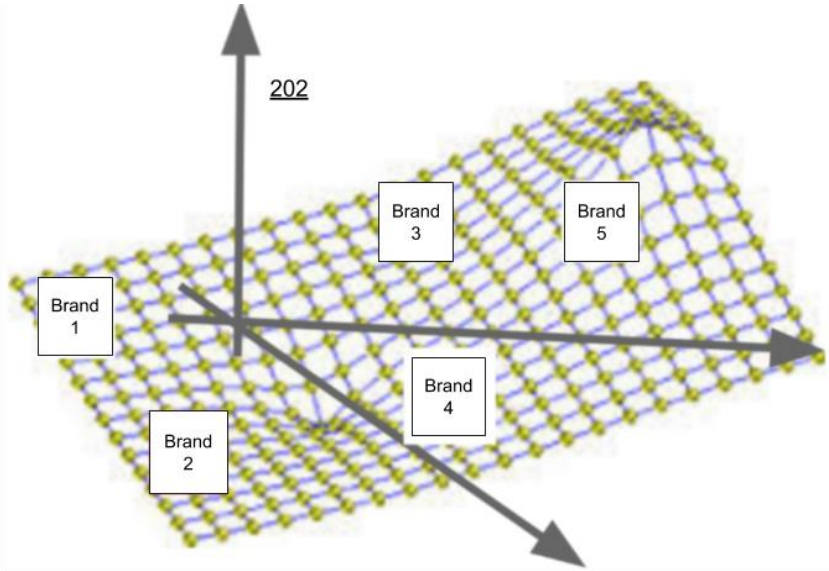


Fig. 2: Embeddings of genuine and phished emails associated with brands cluster in feature space

As explained before, the problem of generating an embedding for visualization of emails (branded and others) is similar to an embedding for face images of humans. In the context of phishing emails, the phished versions (similar to an email) are equivalent to slight modifications of the faces (of a known face). Under a set of reasonable tweaks or variations of the visual and textual (location, font size, etc.) contents of a particular branded email, their renderings cluster tightly around the rendering of the branded email in feature space. This is illustrated in Figure 2, where embeddings of emails (both genuine and phished) of a number of brands form brand-clusters (brand 1, … , brand 5) in feature space (202), which is typically an *N*-dimensional vector space. As shown in Fig. 2, the manifold discovered for the domain of email renderings ensures that genuine and phished email from a given brand lie close to each other, and far away from other emails.

Once the embedding is discovered, the problem of phishing detection for a specific branded email is equivalent to finding the Euclidean distance between the embeddings of a rendering of a phished email and of the rendering of a genuine branded email. A threshold is

applied to declare it as a match or not. The threshold is decided based on the receiver operating curve (ROC) found over a validation dataset.

Training the neural network

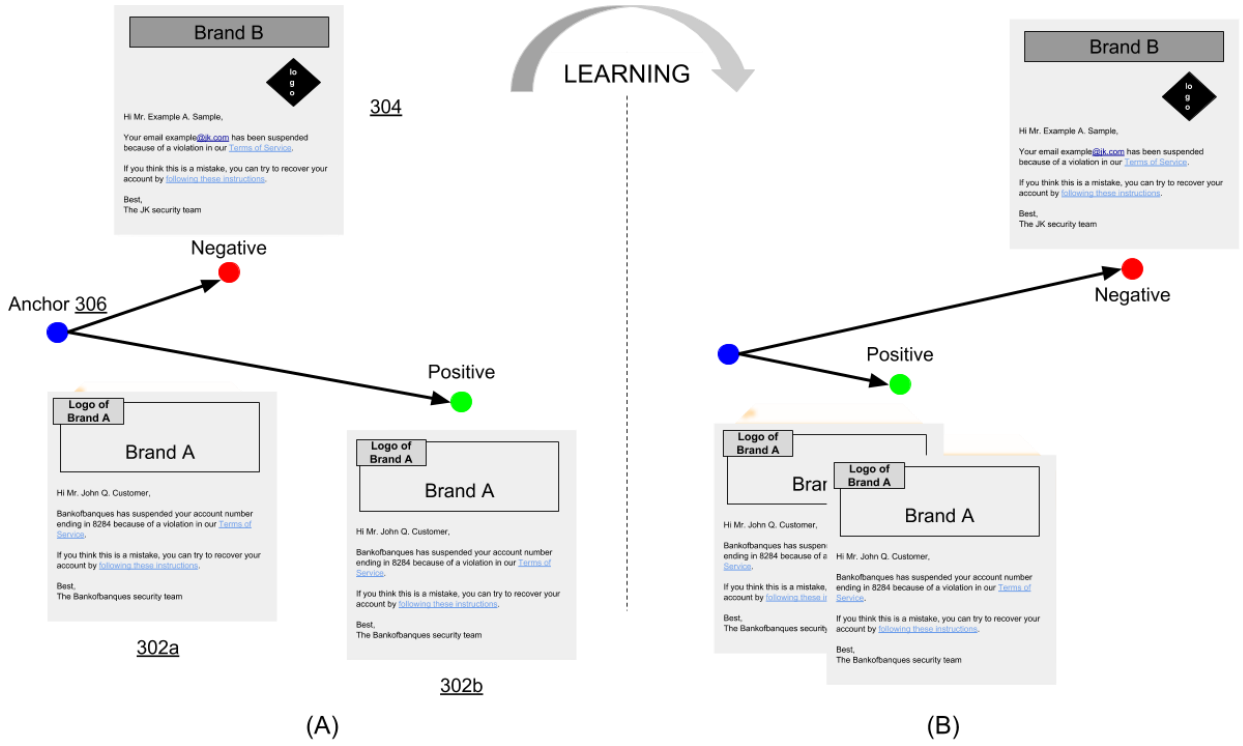


Fig. 3: Training to bring positive examples closer to an anchor while moving negative examples away

Fig. 3 illustrates training of the network to cluster emails of a given brand together. As shown in Fig. 3A, positive examples of branded email (302a-b), including genuine and phished email of that brand, are moved during training towards an anchor (306). Negative examples (304), e.g., genuine or phished emails of another brand, are moved away from the anchor. As a consequence, at the end of training (Fig. 3B), positive brand examples are close to each other, and far away from negative brand examples.

The training process illustrated in Fig. 3 is enabled by optimizing a triplet loss function. The triplet loss function is a ranking loss that, when optimized, clusters images (e.g., their embedding in a Euclidean L2-normed space) belonging to the same label together, while sending images belonging to a different label further away. A triplet consist of an anchor (A), a positive (P) and a negative (N). The loss is defined as follows:

$$L(A, P, N, f) = \left| ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \gamma \right| +$$

where $|.|+$ denotes the hinge loss, $f(.)$ is the to-be-learned projection function from an image to an embedding, and $\gamma$is a margin (gap) parameter. The projection function in this case is a deep network, such as the one in Fig. 1.

Sparsity of training data

Training examples are identified using manual markings and tags. Typically, deep neural networks such as inception require training examples in excess of what is available with manual markings alone. This sparsity of training examples is addressed as follows:

- **Pre-train the network using phished landing pages as examples:** Phished landing-page examples outnumber phished email examples. Since images of rendered landing pages are almost of the same genre as that of rendered emails, such pre-training results in a good starting point, and requires fewer examples for fine tuning as compared to training a network from scratch.
- **Use of data augmentation methods:** Given a branded email, several different versions of the email are generated by tweaking the background color, sub-background sizes, text positions, font-sizes, etc. Figure 4 shows examples of phished training examples generated automatically from a genuine branded training example using, e.g., Monte Carlo technique.

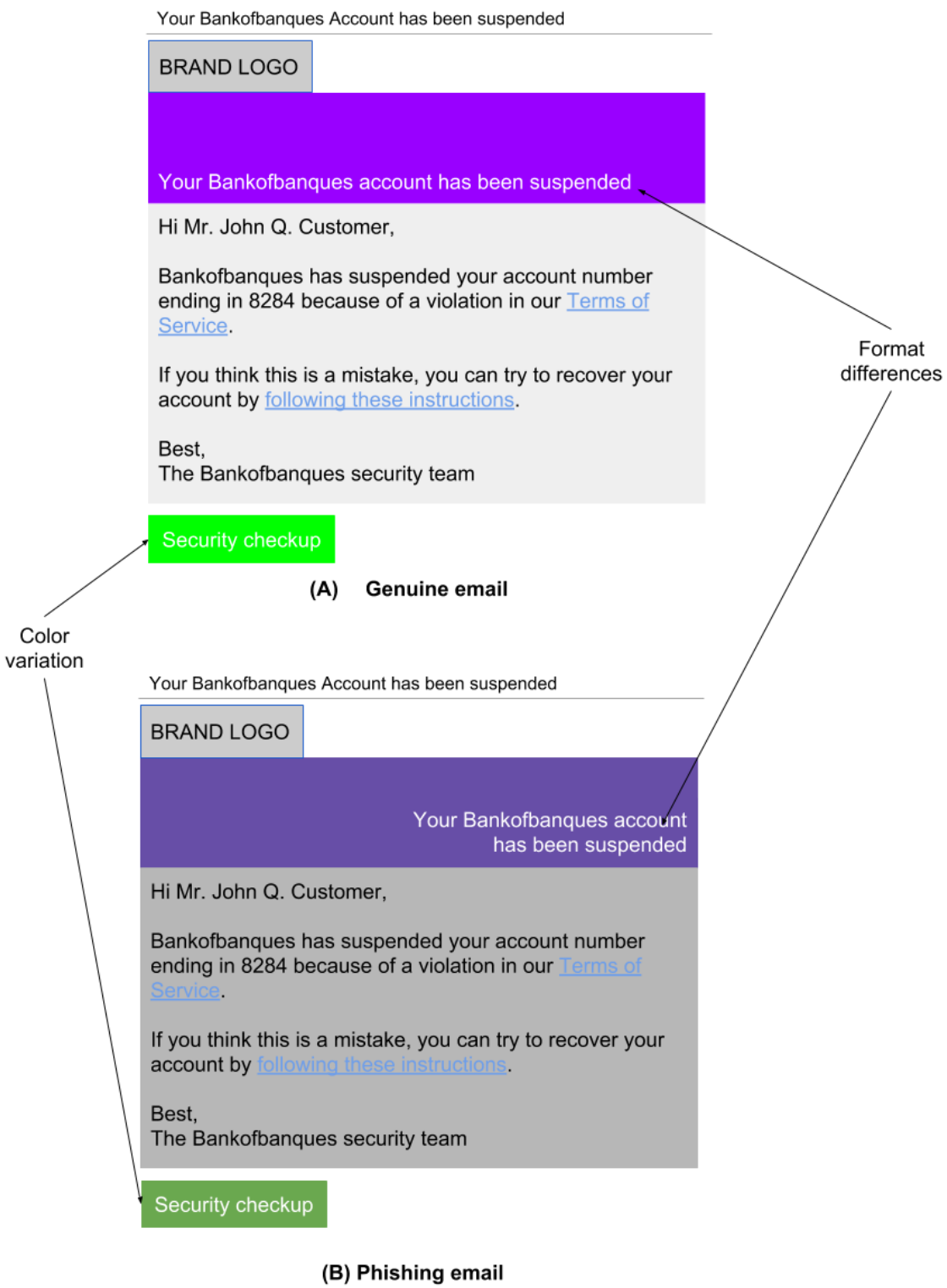
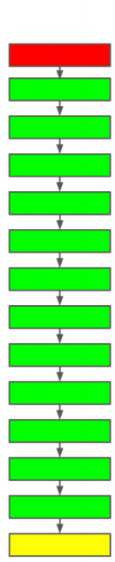


Fig. 4: (A) Genuine email. (B) Phishing email generated from genuine branded email for the purposes of augmenting training data



Type	Output Depth	Output Resolution
3x3 Convolution	32	112
3x3 Separable Convolution	64	112
3x3 Separable Convolution	128	56
3x3 Separable Convolution	128	56
3x3 Separable Convolution	256	28
3x3 Separable Convolution	256	28
3x3 Separable Convolution	512	14
3x3 Separable Convolution	512	14
3x3 Separable Convolution	512	14
3x3 Separable Convolution	512	14
3x3 Separable Convolution	512	14
3x3 Separable Convolution	512	14
3x3 Separable Convolution	1024	7
3x3 Separable Convolution	1024	7
Avg Pool + FC	1000	1

Fig. 5: A mobilenet architecture

Alternative to the machine-learning architectures described above, the mobilenet neural network can also be used. An example mobilenet architecture is shown in Fig. 5. Mobilenets are efficient computer vision models that can be customized to maximize on-device resources. Mobilenets are based on a simple, efficient architecture using depthwise convolutions. Note that 95% of the operations in a mobilenet are 1x1 convolutions, thereby enabling efficient implementation in general matrix-matrix multiplication processors (GEMMs). A simple way to use mobilenet is to integrate its body into the embedding trainer in place of the inception network.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed.

For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure applies machine-learning based techniques to assess the visual similarity of genuine and phished emails (or landing pages) for a given brand. The techniques detect visual near-duplicates of a trusted entity's email and thereby achieve resilience against adversarial attacks. The need for use of hand-crafted features to achieve visual-similarity match is eliminated, enabling accurate detection of new genres of phishing email as they surface.