NPS-AntiClone: Identity Cloning Detection based on Non-Privacy-Sensitive User Profile Data

Ahmed Alharbi, Hai Dong, Xun Yi and Prabath Abeysekara
School of Computing Technologies, Centre for Cyber Security Research and Innovation, RMIT University, Melbourne, Australia
Email: s3633361@student.rmit.edu.au, hai.dong@rmit.edu.au, xun.yi@rmit.edu.au, s3693452@student.rmit.edu.au

Abstract—Social sensing is a paradigm that allows crowdsourcing data from humans and devices. This sensed data (e.g. social network posts) can be hosted in social-sensor clouds (i.e. social networks) and delivered as social-sensor cloud services (SocSen services). These services can be identified by their providers’ social network accounts. Attackers intrude social-sensor clouds by cloning SocSen service providers’ user profiles to deceive social-sensor cloud users. We propose a novel unsupervised SocSen service provider identity cloning detection approach, NPS-AntiClone, to prevent the detrimental outcomes caused by such identity deception. This approach leverages non-privacy-sensitive user profile data gathered from social networks to perform cloned identity detection. It consists of three main components: 1) a multi-view account representation model, 2) an embedding learning model and 3) a prediction model. The multi-view account representation model forms three different views for a given identity, namely a post view, a network view and a profile attribute view. The embedding learning model learns a single embedding from the generated multi-view representation using Weighted Generalized Canonical Correlation Analysis. Finally, NPS-AntiClone calculates the cosine similarity between two accounts’ embedding to predict whether these two accounts contain a cloned account and its victim. We evaluated our proposed approach using a real-world dataset. The results showed that NPS-AntiClone significantly outperforms the existing state-of-the-art identity cloning detection techniques and machine learning approaches.

Keywords—Social-sensor cloud service provider; Identity cloning detection; Non-privacy-sensitive user features; Social media/networks.

I. INTRODUCTION

Social sensing is a paradigm that allows crowdsourcing data from multiple social-sensors such as humans, smartphones, smart glasses, etc. [1]. This sensed data, often termed as social-sensor data, is hosted in social-sensor clouds (i.e. social networks such as Twitter and Facebook), and can take different shapes [2, 3]. Examples of social-sensor data include Facebook status messages and Twitter posts. Social-sensor clouds have, nowadays, become a significant accessible channel to express opinions regarding events and activities [2]. Posts may include critical information, particularly images and descriptions regarding incidents or public events [1].

Nowadays, social-sensor clouds play a very significant role in special events (e.g. sports, crimes, etc.). Social-sensors can publish thousands, or even millions of posts (in the form of texts and/or images) over social-sensor clouds. This large amount of information can be abstracted as social-sensor cloud services (abbreviated as SocSen services) for the convenience of information deployment and reuse [2, 3]. The aforementioned special events can be described from different perspectives, such as where, when, and what through the functional and non-functional properties of SocSen services [4].

The prevalence of social-sensor clouds has attracted many adversaries attempting to exploit SocSen service providers’ identities and deceive users in many ways. Identity cloning is one such mode of deception whereby an attacker registers a fake profile of a provider using the provider’s identity information to perform malicious activities [5]. Examples for such malicious activities include tarnishing the provider’s reputation or deceiving the provider’s friends to steal sensitive information that is not publicly available [5]. For example, it has been reported that the Facebook account of Mark Zuckerberg, Chief Executive Officer of Facebook, has previously been cloned and involved in financial fraud1. Another well-known example is that Russian President Vladimir Putin’s Twitter account was cloned and attracted over 1 million followers2. In addition, these fake identities can also be used to spread fake and false information sourced from unreliable sources rapidly over social-sensor clouds accelerated by the proliferation of the Internet and prevalence of social networks. This can negatively impact thousands of individuals since most individuals tend to believe the information published on social media in regards

---

1https://www.nytimes.com/2018/04/25/technology/fake-mark-zuckerberg-facebook.html
2https://www.abc.net.au/news/2018-11-29/twitter-suspends-account-impersonating-vladimir-putin/10569064
to public events [3].

Identity cloning can take two key shapes in the form of 1) single-site and 2) cross-site identity cloning [5]. The former refers to instances where an attacker creates a cloned identity of a SocSen service provider in the same social-sensor cloud. The latter represents instances where an attacker clones the identity of a provider from another cloud. In this work, we primarily focus on detecting single-site identity cloning.

Most social-sensor clouds lack automated support for detecting identity cloning. For example, Twitter and Instagram review identity cloning claims upon receiving a valid identity cloning report from the end-users. Currently, these platforms do not have support for automated tools to detect identity cloning. Existing research on cloned identity detection mostly relies on both privacy-sensitive user profile data and non-privacy-sensitive user profile data. Privacy-sensitive user profile data, such as full name, date of birth, personal photographs of a user, cannot be accessed by third party applications via Application Programming Interfaces (APIs) or other means due to privacy restrictions. As a result, the existing methods [6], [7], [8] are inherently less applicable for third-party applications to detect cloned identities. Assume that an attacker uses a cloned account to log onto a third party website or application. This third party would find it challenging to accurately identify whether or not it is a cloned account using the existing techniques. In contrast, non-privacy-sensitive user profile data, such as screen name, profile description, etc. of the users, is often readily accessible to third party applications, and can be directly obtained using APIs exposed by social media platforms. Hence, there is an urgent need and potential for exploring approaches to detect identity cloning by utilizing only non-privacy-sensitive user profile data.

Furthermore, most existing identity cloning approaches employ simple feature similarity or supervised machine learning to detect cloned identities [6], [7], [8], [9]. Simple feature similarity usually employs human-defined metrics to calculate profile attributes’ similarities such as TF-IDF based cosine similarity or Jaro–Winkler distance [7], [9], [10]. These metrics are unable to capture the semantics of various literal strings. These metrics focus only on characters distance or word frequency. For example, the aforementioned metrics cannot measure the semantics between the words king and man, which describe a gender-based relationship. On the other hand, supervised machine learned-based approaches require labelled data samples for predictive model training. However, social networks such as Twitter have hundreds of millions of active users. Therefore, it is time-consuming and needs high labour cost to obtain enough labelled data samples to train machine learning models [9].

Different from the existing works, we focus on learning social network users’ multi-view representations. Multi-view learning aims to learn a single function to model multiple views and jointly optimize all the functions to improve their generalization performance [11]. In the context of social network identity, users can be represented by information from multiple views including users’ posts, social relations (e.g. friends and followers) and users’ attributes (e.g. friends count, posts count). The multi-view information comprehensively describes the users and any loss of this information might potentially lead to missing important information [11]. Existing works had only used part of the multi-view information discussed above, thereby leading to unsatisfactory performance [7], [8]. The challenge of learning the user’s multi-view representation lies in the high non-linearity and high-dimensional characteristics of the social media data, such as users’ posts and social relations networks [12]. Therefore, there exists an urgent need for alternative approaches with a strong ability to learn features that can more accurately represent the multi-view characteristics of the SocSen service providers.

To address the above limitations, we propose a novel unsupervised approach, NPS-AntiClone, for SocSen service providers’ identity cloning detection. NPS-AntiClone consists of three main components, namely, 1) a multi-view account representation model, 2) an embedding learning model and 3) a prediction model. The multi-view account representation model generates three different views from the non-privacy-sensitive user profile data of a pair of accounts sharing the same screen name or username. These views can represent a user’s comprehensive social network features, which are highly likely to be mimicked by attackers. The generated views include a post view, a network view, and a profile attribute view. In the post view, we extract a pre-trained language representation using Sentence-BERT (SBERT), which is a model trained on top of a large amount of unannotated data (e.g. a Wikipedia dump). It aims to extract a textual representation from a given text content and transform it into a form where similarity measures can be applied. For the network view, we consider two types of account networks: follower and friend networks, and then we learn the respective network representations using Node2vec [13]. Meanwhile, for the profile attribute view, we employ 12 public profile attributes. The embedding learning model adopts weighted generalized canonical correlation analysis (wGCCA), which learns a single embedding from the generated multi-view representation for each account [14]. It then calculates the cosine similarity between the account pair to predict whether or not the pair of accounts contains a cloned account and its victim based on their similarity. Our proposed approach shows promising performance in the

---

1[https://help.twitter.com/en/rules-and-policies/twitter-impersonation-policy]
2[https://help.instagram.com/446663175382270]
evaluation based on a real-world Twitter dataset. Our main contributions can be summarized as follows:

- We designed a novel unsupervised SocSen service providers’ identity cloning detection approach for third party applications/websites, which depends only on non-privacy-sensitive user profile data accessed via social-sensor cloud APIs.
- We devised an approach to automatically generate a multi-view representation that can comprehensively describe an account from a post view, a network view, and a profile attribute view.
- We adopted a technique that effectively learns a single account embedding from all the multi-view account representations, which is found to be more robust.
- We present an exhaustive evaluation carried out atop a real-world dataset we collected from publicly available Twitter data. We also show that NPS-AntiClone outperforms the state-of-the-art identity cloning detection approaches and other machine learning based approaches using non-privacy-sensitive user profile data.

The rest of the paper is structured as follows. Section II reviews existing SocSen service studies and state-of-the-art identity cloning detection techniques. Section III presents the details of our proposed approach. Section IV describes the evaluation process of our proposed approach. Section V concludes the paper.

II. RELATED WORK

In this section, we reviewed and analyzed the related work in SocSen services and identity cloning detection.

A. Social-Sensor Cloud Services

SocSen services employ a service paradigm to abstract social-sensor data into small independent functions. Each SocSen service is defined by its functional and non-functional properties. These functional properties include descriptions, tags, etc. embedded in social-sensor data. The non-functional properties contain spatio-temporal and contextual information, as well as qualitative features (e.g. trust, price, etc.) of social-sensor data. SocSen services aim to reduce social-sensor data complexity and also support real-time and efficient access to relevant and high-quality data.

SocSen services provide an easy way to access and manage social-sensor cloud data for building the applications of scene analysis. Aamir et al. proposed a set of SocSen service selection frameworks to select scene-related social media images based on user queries. Social media images are abstracted as SocSen services. These frameworks try to match SocSen services and user queries in terms of their spatio-temporal and contextual correlations. Aamir et al. further proposed a series of SocSen service composition models for scene analysis. These models take into account the spatio-temporal, image metadata and textual features (i.e. comments and descriptions) of the SocSen services to reconstruct tapestry scenes.

B. Identity Cloning Detection Techniques

A significant number of approaches had been proposed in the current literature to detect fake identities or spammers on social media. The most commonly used approaches promote creating behavioural profiles to distinguish between trustworthy and untrustworthy users. These behavioural profiles include characteristics of users such as writing style or following accounts, etc. However, behavioural profiles features are not optimal to detect identity cloning, since an attacker trying to clone the identities can mimic their profile attributes. Therefore, we need to employ features that are able to accurately characterize account pairs to detect the cloned identity. Other works use the trust relationship between users in social networks. The main assumption of these techniques is that a fake/spammer account cannot build an arbitrary number of trusted connections with legitimate accounts in social networks. This assumption might not hold true when dealing with cloned accounts. In the context of cloning accounts, attackers attempt to mimic legitimate accounts. Thus, it is much easier for cloned accounts to build connections with the legitimate accounts than other types of fake identities.

A few approaches had been proposed in the current literature to detect identity cloning on social media. Kontaxis et al. proposed a methodology that allows users to see if they have fallen victim to identity cloning. Devmane and Rana designed a methodology to detect identity cloning attacks in both single and cross-site contexts. It searches for similar user profiles and then calculates a similarity index to detect the cloned profiles. Jin et al. analysed and characterized the behaviours of the identity cloning attacks. They proposed two profile similarity schemes to detect suspicious profiles. Kamhoua et al. overcame identity cloning attacks by matching user profiles across multiple social media. They used a hybrid string-matching similarity algorithm to find profile similarity. Goga et al. proposed an approach that can detect impersonation attacks. Their approach detects whether a pair of profiles are controlled by the same person or imposter. It compares the impersonation account activity and reputations. It leverages a Support Vector Machine (SVM) classifier to detect impersonation attacks.

Most of the existing studies detect identity cloning based on both privacy-sensitive and non-privacy-sensitive user profile data. However, many third party applications or websites that employ social media identities for authentication are not able to access the privacy-sensitive user profile data using social media APIs. Therefore, the performance of these approaches is questionable for these third parties. In addition, most of the existing approaches are built on simple feature similarity models or classic supervised machine learning
models. Instead, we propose an unsupervised approach that is able to learn the account multi-view and effectively fuse them to detect identity cloning.

III. PROPOSED SOLUTION

In this section, we present a detailed overview of the proposed NPS-AntiClone approach and its key components.

A. Overview

Given a set of social media accounts, we aim to detect identity cloning using multi-view representation of an account. Firstly, we generate different views based on 1) posts, 2) network and 3) profile attribute of a given social media account. For the post view, we use Sentence-BERT (SBERT) to extract the account’s post representation. For the network view, we build two networks, namely, followers and friends and then extract the network representation using Node2vec, which is an algorithmic framework for graphic representation learning. For the profile attribute view, we extract 12 public profile attributes from the account public profile. We then use wGCCA to learn a single embedding from all these views. We then calculate the cosine similarity between an account pair to predict whether or not it contains an account and its replica. Figure 1 shows the workflow of NPS-AntiClone.

Suspicious Account Pair Generation: Social media networks typically contain a very large number of users. Therefore, we need to generate suspicious account pairs for identity cloning detection. Here we adopted a mechanism commonly used in existing works [9], [20], whereby we search for a pair of accounts that contain a similar username or screen name. For each suspicious account pair, we generate multi-view account representation. If a match is found between the two accounts in an account pair, it implies that the two accounts have a potential likelihood to represent the same individual but their true labels are unknown.

B. Multi-view Account Representation Model

Our goal is to generate a multi-view account representation for a given user account by combining several views that correspond to the non-privacy-sensitive profile data of the account. We use three different views based on the account’s 1) post view, 2) network view and 3) profile attribute view. These views can comprehensively represent a user account, which are highly likely to be mimicked by attackers. Then, wGCCA is used to learn a single embedding from these views [21]. We describe each view in detail in the following subsections.

1) Post View: To construct the post view, we extract a pre-trained language representation for each account. In this paper, we use the Sentence-BERT (SBERT) [22] to generate the vector-space representations of the account posts. SBERT is an adjustment of the pre-trained bidirectional encoder representations from transformers network (BERT) [23]. BERT is a technique for pre-training language representations. It can be used for extracting high-quality language features or can fine-tune these models on a downstream task such as classification, question answering, etc. [23]. These pre-trained models are very efficient in extracting the text representation for any specific task from the transformer architecture [24]. SBERT uses siamese and triplet network structures to obtain semantically meaningful sentence representation. Models trained using SBERT are also based on transformer networks similar to the BERT models. In addition, SBERT also adds a pooling operation to the output of BERT to obtain a fixed-sized sentence representation. The sentence representation is usually obtained by computing the mean of all output vectors. We gathered
n posts publicly available in a given account, denoted as $T = (t_1, ..., t_n)$ to obtain the posts’ representation for a user $u$. The representation of each post $t_i (i \in 1, .., n)$ is obtained from the pre-trained language representation.

Figure 2 shows the architecture of SBERT. Each post $t_i$ is first tokenized into a single word $w_i$ and then a special token named [CLS] is added to mark the beginning of a sentence. [SEP] is also a special token added to mark the end of the sentence. Then, a set of tokenized words is passed through BERT to embed fixed-sized sentences. Then, in the pooling layer, a mean aggregation (see Equation 1) is applied to generate $t$ representations. The mean aggregation is known to have higher performance compared to CLS aggregation or max [22]. The output of each post is made up of 768 dimensions, which are the default setting of BERT. Finally, for each post in an account, we compute the mean of all the posts’ representation $T$ to get an aggregated representation, as below.

$$t = \frac{1}{|T|} \sum_{t_i \in T} w_i \quad \text{(1)}$$

2) Network View: A network of accounts is a set of users that engage in various kinds of interactions, such as friendship, retweet, etc. within a social network, which can be presented as a networked graph. If an account in this network interacts with another account, there will be an edge between them in the corresponding networked graph. We consider two types of account networks: 1) follower network: In this network, if an account follows a specific user such as a celebrity or a friend, there will be an edge between them, and 2) friend network: In this network, if an account gets a follow by another account, there will be an edge between them. Inspired by the success of the graph representation, we utilize the Node2Vec to learn the network representation, or in other words, network view of an account [13]. Node2vec is a broadly applied unsupervised representational learning approach for graphs. It tries to maximize the log-probability between neighbours of a node, or in other words, accounts with an edge between them, via a biased random walk method. Node2vec will generate sentences from the graph. A sentence is a list of nodes (directed subgraphs) that makes a corpus. Then, these corpora are feed into a skip-gram model to obtain node feature representation [23]. Skip-gram is an unsupervised learning approach that is applied to obtain the most related words for a given word. It is also applied to predict the context word for a given target word.

Let us denote an account’s follower network as $G(V, E)$, where $V = [v_1, v_2, ..., v_n]$ represents a set of nodes (i.e. users), $n$ represents the total number of nodes and $E$ represents a set of edges (i.e. social interactions). For every node $v \in V$, let $N_u(v) \subset V$ represent a network neighborhood of node $v$ that is constructed over a neighbourhood sampling strategy $S$. $S$ is a flexible biased random walk technique that can trade off between local and global views of the network in a depth-first (DFS) as well as breadth-first (BFS) fashion. Node2Vec can learn the node’s representation by optimizing the objective function as follows:

$$\max_f \sum_{v \in V} \log Pr(N_S(v) | f(v)) \quad \text{(2)}$$

where $f : V \rightarrow \mathbb{R}^d$ represents a mapping function that aims to learn mapping nodes to feature representation ($V \in \mathbb{R}^d$). Node2Vec uses stochastic gradient descent (SGD) as its underlying optimization method to learn the node representation mapping function $f()$. We utilize Node2vec to generate the follower network and friend network of the account since Node2vec can learn the high-level feature representations that include deep social interaction of accounts. This information can be helpful for differentiating between an account pair. We assume that it is hard for an attacker to mimic these deep social interactions.

3) Profile Attribute View: To construct the profile attribute view of an account, we extract 12 public profile attributes of an account to create an attribute vector. These attributes can categorize the activities and reputation of an account. For example, the number of tweets can indicate the activities of an account, while the number of followers can indicate the reputation of an account. We consider the following profile attributes:

1) Friend (following) count: The number of users that the account follows.
2) Follower count: The number of followers that the account has.
3) Favorite count: The number of tweets that the account has liked.
4) Tweet count: The number of tweets (including retweets) that the account has posted.
5) List count: The number of public lists of which the account is a member.
6) Account age: The life time of the account to-date in months from the date of the account registration.
7) Profile background: A binary value that shows whether the account has not altered the background or the theme of their profile.
8) Profile image: A binary value that shows whether the account has not uploaded their profile image and that a default image is used.
9) Has profile description: A binary value that shows whether the account has added a description to their profile.
10) Profile URL: A binary value that shows whether the account has added a URL to their profile.
11) Screen name length: The length of the screen name of the account.
12) Description length: The length of the description of the account.

Figure 3 shows the visualization of all proposed views using t-Distributed Stochastic Neighbor Embedding (t-SNE) on each view vector. t-SNE is an unsupervised, non-linear approach predominantly utilised to ease data visualization and reduce dimensionality. It provides an intuitive view of how the data is arranged in a high-dimensional space. Cloned accounts and non-cloned accounts are represented by the red and black circles, respectively. It can be inferred that each of the proposed views can be employed to calculate the profile similarity between the accounts and that we can get a better profile similarity performance by effectively aggregating the proposed views.

C. Embedding Learning Model

The views described in the previous section includes some knowledge that might be utilised to detect cloned accounts. Employing a single view might result in the loss of valuable information. A simple and naive technique is to consolidate all these views by concatenating them together. However, a drawback of this concatenation is that it causes over-fitting on a small training sample because of the large account representation. It might also disregard the meaningful knowledge included in the views because each view has a specific statistical property. Therefore, we use an alternate technique named CCA to learn a comprehensive single embedding from the proposed views. CCA finds maximal information from two sets of views and generates an embedding. CCA has been applied to several multi-view data learning problems and achieved successes. The limitation of CCA is that it can only learn two views. Therefore, we employ Generalized CCA (GCCA) that allows multi-view based learning. GCCA contains many variations, such as [28], [29], [30]. Here we employ Carroll’s GCCA, since it is based on an eigenequation that is computationally straightforward and efficient. The GCCA objective can be formulated as follows:

\[
\arg \min_{G, U_i} \sum_i \| G - X_i U_i \|^2_2 \quad \text{s.t.} \quad G^T G = I
\] (3)

where \( X_i \in \mathbb{R}^{n \times d_i} \) corresponds to the data matrix of the \( i^{th} \) view, \( G \in \mathbb{R}^{n \times k} \) contains all learned account embedding and \( U_i \in \mathbb{R}^{d_i \times k} \) maps from the latent space to the observed view \( i \). However, each view can be more or less important for detecting identity cloning. Therefore, we use weighted GCCA (wGCCA) [14]. wGCCA adds weight \( w_i \) for each view \( i \) to Equation [3] as follows:

\[
\arg \min_{G, U_i} \sum_i w_i \| G - X_i U_i \|^2_2 \quad \text{s.t.} \quad G^T G = I, w_i \geq 0
\] (4)

where \( w_i \) represent the weight of a view and this weight shows the view’s importance. The columns of \( G \) are the eigenvectors of \( \sum_i w_i X_i (X_i^T X_i)^{-1} X_i^T \) and the solution for \( U_i = (X_i^T X_i)^{-1} X_i^T G \).

D. Prediction Model

We used the cosine similarity to compute the similarity between the account pair as follows:

\[
\cos(A_1, A_2) = \frac{A_1 \cdot A_2}{||A_1|| \cdot ||A_2||}
\] (5)

where \( A_1 \) and \( A_2 \) refer to the account pair embedding. A threshold will be selected to determine whether a given account pair contains a cloned account and its victim based on their similarity score.

IV. Evaluation

A set of experiments was performed to evaluate and analyse the effectiveness of NPS-AntiClone against existing state-of-the-art identity cloning detection approaches. In addition, we also evaluated several candidate machine learning models to assess their performance in this particular problem context. All the experiments were conducted on a computer with Intel Core i5 1.80 GHz CPU and 16 GB RAM. In addition, all the candidate models compared were implemented in Python. We used the SBERT package to extract the pre-trained language representations used by NPS-AntiClone. We also used the StellarGraph package to extract the Node2vec representations used by NPS-AntiClone. We used Tensorflow (v1.14.0) library written in Python to implement the deep learning models evaluated and scikit-learn (v0.22.1) for the other machine learning models evaluated. All the machine learning models were run for 10 rounds with different random permutations of the data. The results were presented as an average computed across all rounds of the experiments, together with standard deviation.

\[\text{https://github.com/UKPLab/sentence-transformers}\]
\[\text{https://github.com/stellargraph/stellargraph}\]
\[\text{https://github.com/UKPLab/sentence-transformers}\]
\[\text{https://www.tensorflow.org/}\]
\[\text{https://scikit-learn.org/stable/}\]
Table I: Statistics of the dataset

| Dataset | #Accounts | #Tweets   | #Friend network | #Follower network |
|---------|-----------|-----------|-----------------|------------------|
| Twitter | 4030      | 948,144   | 514,534         | 365,687          |

A. Dataset

To the best of our knowledge, there are no readily and publicly available datasets that could be used for evaluating identity cloning in the domain of social networks. Most existing works, albeit limited, evaluated their proposed approaches on simulated data. Therefore, to evaluate NPS-AntiClone, we developed a dataset via authorized non-privacy sensitive user profile data fetched from Twitter APIs. We collected 4,030 public Twitter accounts (2,015 cloned accounts and their corresponding victim accounts). Table I summarizes the detailed statistics. We also randomly collected 20,152 non-cloned public Twitter accounts to add noises to the dataset. In total, we have 24,182 public Twitter accounts. The dataset was randomly split with a 80:20 training-to-split ratio in order to derive training and test datasets for all the supervised machine learning models.

B. Other Approaches Evaluated

We compared and evaluated NPS-AntiClone against the following approaches which include a set of state-of-the-art identity cloning detection approaches, supervised machine learning approaches and variants of NPS-AntiClone:

- **Basic Profile Similarity (BPS)** [10]: This technique calculates the number of similar public attributes and common friends between a given user profile and its suspected cloned account(s). Our experiments only used the number of common friends in the friend list in the compared accounts since the recommended and excluded friend lists are privacy-sensitive user profile that is not available publicly. Further, in our dataset, each account has 13 public attributes. We also set $\mu = 0.0154$, $\varepsilon = 13$, and $\lambda = 0.03$. If the profile similarity of a given pair of accounts is less than $\mu$, then they are considered as non-cloned. For a given pair of accounts, $\mu$ is calculated based on the smallest number of the attribute similarity $a$ of the account pair and the smallest number of common friends $b$ of the account pair (see Equation 6). $\varepsilon$ is the number of public attributes (raw data) in our dataset. $\lambda$ is calculated based on the smallest $a$ and largest $b$ number of common friends in the account pair, respectively (see Equation 6).

$$
\mu, \lambda = \frac{(a)^2 + (b)^2}{\sqrt{k^2 + x^2}}
$$

where $k$ and $x$ are adjusted based on the contribution of similar public attributes and the number of common friends. We assigned 0.5 for $k$ and $x$ to make them contribute equally.

- **Devmane and Rana** [7]: This technique compares names, education, profile photos, places lived, birth date, workplace, gender, photos added to the profile, number of friends/connections. We only compared the screen name, places lived (location) and the number of friends/connections that can be fetched using Twitter APIs. We then calculated a similarity index for a given pair of accounts. The original work does not show adequate details on the type of similarity index they used in their proposed approach. Therefore, we adopted the screen name and location similarity technique used in our work to calculate the similarity index of the compared user profiles.

- **Goga et al.** [9]: This technique compares the time at which the pair of accounts were created and the reputations of the accounts in terms of popularity and
social influence, and detects whether an account pair has an impersonated account. To determine this, it uses four different types of features, namely, profile similarity, social neighbourhood overlap, time overlap accounts and differences between accounts. It then trains an SVM classifier, with a linear kernel to classify if a given account is impersonated. In our experiments, we used all the recommended features in the original work to train the SVM classifier.

- **Kamhoua et al. [8]**: This technique compares friends list similarity and calculates attribute similarity using a similarity metric called Fuzzy-Sim. To calculate the attribute similarity, it considers the following attributes: name, education, city, age, workplace, gender, and friend list. In our experiments, however, we only compared the screen name, city (location), and the friend list provisioned by Twitter APIs. We then calculated the attribute similarities and friend list similarity using FuzzySim with the same threshold values recommended by the original work (0.565 and 0.575).

- **Zheng et al. [18]**: This is a typical spammer detection model. It employs 18 features, which are a combination of profile-based features such as the number of followers, the number of followers, and content-based features such as the average number of hashtags. It then trains an SVM classifier, with a Radial Basis Function (RBF) kernel to classify an account as a spammer or non-spammer account. In our experiments, we used all the recommended features recommended in the original work to train the SVM classifier.

- **Other Supervised Machine Learning Models**: We use the following supervised machine learning and deep learning models in order to predict whether a given account pair form a cloned and non-cloned pair. These models are Adaboost (ADA), Convolutional Neural Network (CNN), Deep Neural network (DNN), K nearest neighbours (KNN), Logistic Regression (LR), Multi-layer Perceptron (MLP), Random forest (RF) and SVM with a linear kernel. Instead of calculating the cosine similarity between the account pair using Equation 5, we feed the classifier with the wGCCA output to predict whether the account pair consists of a cloned account and its victim.

- **NPS-AntiClone**:

  - **NPS-AntiClone**$_{concat}$: In this approach, we concatenate all the views in order to combine them into a single account representation of the size $n = 1,036$.

  - **NPS-AntiClone**$_{posts}$: This approach is a variant of NPS-AntiClone that only uses the post view of the accounts to detect an account pair with a cloned account and its victim.

  - **NPS-AntiClone**$_{net(F, FL)}$: This is also a variant of NPS-AntiClone that only employs the network’s view representation. Additionally, we attempt to use the follower network representation and friend network representation separately, and denote them as NPS-AntiClone$_{F, L}$ and NPS-AntiClone$_{F}$ respectively.

- **NPS-AntiClone**$_{PA}$: This is a variant of NPS-AntiClone that only employs the profile attribute view of the accounts.

### C. Hyperparameter Tuning

All the hyperparameters of the supervised machine learning models were properly tuned to obtain their optimal performance. Table II shows the hyperparameter values used for the machine learning and deep learning algorithms evaluated. We set the parameters of the state-of-the-art approaches following the original works.

For the proposed NPS-AntiClone, the suspicious account pairs were selected only if the similarity score of the screen names or usernames between a pair of accounts is over 80%, by following the existing work [9]. We used 'paraphrase-distilroberta-base-v1' as the pre-trained model for SBERT. This pre-trained model was trained on millions of paraphrase sentences. The dimension size of the post representation using SBERT is 768, which is a default setting of the model. The default dimension size of the Node2vec for both the follower network and friend network is 128. We also use the probability for moving away from source node $q = 2$, the probability of returning to source node $p = 0.5$, the number of random walks per root node $n = 10$ and the maximum length of a random walk is 15. All the profile attribute views were normalized to range 0 and 1. The weight $w$ of the wGCCA is set as $[0.25, 0.5, 0.5, 0.25]$. The optimal threshold of a cosine similarity score between two account embedding is 0.1 according to our experiment (described in Section IV.D)

### D. Results and Discussion

We evaluated and compared the performance of all the aforementioned models using Precision, Recall, F1-Score and F2-Score as our Key Performance Indicators (KPIs). Precision measures the percentage of predicted account pairs (a cloned account and its victim) that are correctly detected. Recall measures the percentage of the true account pairs.

---

1. https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v1
that are correctly detected. F1-score is the harmonic mean of precision and recall. F2-score puts more emphasis on identifying as many account pairs with a cloned account in the dataset as possible.

1) Overall performance: The results of the experiments showed that our proposed approach clearly outperforms the existing state-of-the-art identity detection techniques across all the KPIs evaluated (see Table III). Based on the optimal threshold (i.e. 0.1), our proposed approach achieved a Precision of 88.70%, Recall of 82.83%, F1-Score of 85.66%, and F2-Score of 83.94%, respectively, which were 13.58%, 10.29%, 11.86% and 10.9% higher than that of the BPS [10], the second best-performing approach. In addition, our proposed approach also outperformed the other existing models in a recall-prioritized situation. A recall-prioritized scenario where a more emphasis is put on identifying as many account pairs with a cloned account in the dataset as possible. The identity detection approaches proposed by Kamhouna et al. [8] and Devmane and Rana [7] performed poorly because they only focus on a simple technique to compute profile attribute similarity and ignores the network information and contents of the account. BSP [10], on the other hand, depends on the simple profile attributes-based similarity and common friends between the account pair, but it does not consider the effect of the network information and the account’s content. Goga et al. [9] depends on a traditional similarity technique, which is based on human defined metrics, to compare between two accounts. Zheng et al.’s model [18] only focuses on spammer detection. It detects whether an account is a spammer or not by employing a set of features that compares spammer behaviour patterns. Likely, the behaviour of cloned identity patterns differs from spammer behaviour. Cloned identity tries to emulate the legitimate account behaviour and therefore, is hard to detect by the spammer detection techniques. NPS-AntiClone is an unsupervised approach that employs non-privacy-sensitive account multi-view representation. The account multi-view representation can capture both the semantic and literal characteristics of the account, which can be attributed to its superior performance compared to the spammer detection model proposed by Zheng et al.

Figure 4: Impact of the multi-view representation: We also compared NPS-AntiClone with the variants announced in Section IV.C, namely, NPS-AntiClone_{posts}, NPS-AntiClone_{net}, NPS-AntiClone_{net(F,FL)}, NPS-AntiClone_{FL}, NPS-AntiClone_{F}, and NPS-AntiClone_{concat}, and the obtained results are depicted in Figure 4. We observed that NPS-AntiClone_{net(F,FL)} performs better than the other variants, which shows the strength of the network representation when representing the account’s followers and friends. NPS-AntiClone_{net(F,FL)} achieved a F-1 score of 61.81%, which is 11.07% and 8.45% higher than that of the NPS-AntiClone_{FL} and NPS-AntiClone_{F}, respectively. The approach that combines NPS-AntiClone_{net(F,FL)} and NPS-AntiClone_{posts}, and NPS-AntiClone_{net(F,FL)+posts} was observed to outperform all other variants by approximately 18% in terms of the F-1 score. Furthermore, merging the NPS-AntiClone_{FA} with NPS-AntiClone_{net(F,FL)+posts} improved the performance of the final proposed model by 13% in terms of F-1 score.

We also experimented with an alternative way to combine all the views. In that, we concatenated all the views into a single account representation. We found that NPS-AntiClone_{concat} performs poorly compared to the NPS-AntiClone, and that NPS-AntiClone outperforms the NPS-AntiClone_{concat} by almost 20% in terms of the F-1 score. The performance results indicated that the learned account multi-view using wGCCA can benefit the identity cloning detection by improving the performance.

3) Impact of the cosine similarity score: NPS-AntiClone depends on the cosine similarity score to decide whether one of the accounts in a given account pair is cloned or not. Therefore, an important question this raises is how this similarity score impacts the performance of NPS-AntiClone. Figure 5 shows the performance results of NPS-AntiClone based on different values of the cosine similarity score.

Table III: Comparison with state of the art identity cloning approaches.

| Model               | Precision (%) | Recall (%) | F1-Score (%) | F2-Score (%) |
|---------------------|---------------|------------|--------------|--------------|
| BSP [10]            | 75.12         | 72.54      | 73.80        | 73.04        |
| Devmane and Rana [7]| 66.32         | 68.86      | 67.56        | 68.33        |
| Goga et al. [9]     | 63.54         | 79.61      | 70.67        | 75.77        |
| Kamhouna et al. [8] | 67.41         | 71.44      | 69.36        | 70.59        |
| Zheng et al. [18]   | 68.15         | 73.34      | 70.64        | 72.23        |
| NPS-AntiClone       | 88.70         | 82.83      | 85.66        | 83.94        |

Figure 5 shows the performance results of NPS-AntiClone when representing the account’s followers and friends. which shows the strength of the network representation and NPS-AntiClone, when concatenating all the views. In that, we concatenated all the views (F,FL), NPS-AntiClone_{FL}, NPS-AntiClone_{F}, and NPS-AntiClone_{concat}, and the obtained results are depicted in Figure 4. We observed that NPS-AntiClone_{net(F,FL)} performs better than the other variants, which shows the strength of the network representation when representing the account’s followers and friends. NPS-AntiClone_{net(F,FL)} achieved a F-1 score of 61.81%, which is 11.07% and 8.45% higher than that of the NPS-AntiClone_{FL} and NPS-AntiClone_{F}, respectively. The approach that combines NPS-AntiClone_{net(F,FL)} and NPS-AntiClone_{posts}, and NPS-AntiClone_{net(F,FL)+posts} was observed to outperform all other variants by approximately 18% in terms of the F-1 score. Furthermore, merging the NPS-AntiClone_{FA} with NPS-AntiClone_{net(F,FL)+posts} improved the performance of the final proposed model by 13% in terms of F-1 score.

We also experimented with an alternative way to combine all the views. In that, we concatenated all the views into a single account representation. We found that NPS-AntiClone_{concat} performs poorly compared to the NPS-AntiClone, and that NPS-AntiClone outperforms the NPS-AntiClone_{concat} by almost 20% in terms of the F-1 score. The performance results indicated that the learned account multi-view using wGCCA can benefit the identity cloning detection by improving the performance.
Recall F1-Score require any training. However, NPS-AntiClone only calculates the labelled data in order to obtain a satisfactory performance with standard deviation ($\sigma$). Machine learning and deep learning models require more all the supervised models (see Table IV). The supervised els revealed that NPS-AntiClone significantly outperformed and the other machine learning and deep learning models:...

5) Impact of the weight of the wGCCA: We also performed multiple rounds of experiments to study the impact of the weight of the wGCCA on each view. To do that, we tested different combinations of weight (i.e. 0.25, 0.5 and 1) for each view. We assigned a weight for each view [posts, netF, netFL, PA]. Figure 6 shows the top 10 results observed against different combinations of the weights. It can be seen that when the profile attribute view was given a low weight, the F1-Score was increasing. We also noticed that when the netF, netFL were getting a high weight, the F1-Score was increasing. The weight of the posts was not observed to have a high impact on the wGCCA. The optimal vector of weights observed for the features [posts, netF, netFL, PA] was [0.25, 0.5, 0.5, 0.25].

V. CONCLUSION AND FUTURE WORK

We proposed a novel unsupervised SocSen service provider identity cloning detection approach, NPS-AntiClone, based on non-privacy-sensitive user features. We devised a multi-view account representation model that generates different views for each account in an account pair comprising three categories of views, namely, 1) post view, 2) network view and 3) profile attribute view. We then adopted wGCCA to learn a single embedding from the generated multi-view. Finally, we calculate the cosine similarity between the account pair. NPS-AntiClone was evaluated on a real-world Twitter dataset against other state-of-the-art cloned identity detection techniques and supervised machine learning models. The results showed that the proposed approach significantly outperformed the other models.

In future, we plan to design a method that is able to point out which account of an account pair is the cloned identity, since our proposed approach only detects whether an account pair form a pair of cloned accounts and its victim.
