One-class Classification with Deep Autoencoder Neural Networks for Author Verification in Internet Relay Chat

Sicong Shao
NSF Center for Cloud and Autonomic Computing
The University of Arizona,
Tucson, Arizona
sicongshao@email.arizona.edu

Cihan Tunc
NSF Center for Cloud and Autonomic Computing
The University of Arizona,
Tucson, Arizona
cihantunc@email.arizona.edu

Amany Al-Shawi
National Center for Cybersecurity Technology
King Abdulaziz City for Science and Technology,
Riyadh, Saudi Arabia
aalshawi@kacst.edu.sa

Salim Hariri
NSF Center for Cloud and Autonomic Computing
The University of Arizona,
Tucson, Arizona
hariri@email.arizona.edu

***Abstract*—Social networks are highly preferred to express opinions, share information, and communicate with others on arbitrary topics. However, the downside is that many cybercriminals are leveraging social networks for cyber-crime. Internet Relay Chat (IRC) is the important social networks which can grant the anonymity to users by allowing them to connect channels without sign-up process. Therefore, IRC has been the playground of hackers and anonymous users for various operations such as hacking, cracking, and carding. Hence, it is urgent to study effective methods which can identify the authors behind the IRC messages. In this paper, we design an autonomic IRC monitoring system, performing recursive deep learning for classifying threat levels of messages and develop a novel author verification approach with one-class classification with deep autoencoder neural networks. The experimental results show that our approach can successfully perform effective author verification for IRC users.**

***Keywords*— Author verification; deep learning; cybersecurity; autoencoder; Internet Relay Chat (IRC);**

I. Introduction

The ubiquity, dissemination, and potential anonymity of social network services make them ideal platforms for cyber-crime. Cybercriminals can exploit the tremendous power of social networks to perform illegal activities like hacking service trade, drug and weapon marketing, money laundering, and so on [1]. One notorious example is Silk Road black market that provides hacking tools, carding, drugs, and more illegal content as online marketplaces accompanied by discussion forums that use Tor network [22]. Also, a decade ago, an anonymous organization mainly used centralized web forums and printed magazines for the propagation of their opinions and information. However, these are largely replaced by social networks nowadays. Therefore, analyzing social networks has become critically important to cybersecurity research.

Internet Relay Chat (IRC) is a real-time communication method for text-based online chat. IRC has been traditionally utilized for legitimate functions like casual chat and technical support, but hackers (the term hacker is used to represent the malicious users and cyber-attackers throughout this paper) also exploit the tremendous dissemination power of IRC to perform malicious activities such as cyberattack, hacking tool propagation, cracking training, and so on [1], [37]. Some IRC channels even contain underground markets where anonymous users sell stolen credit card data, hacking service, security exploits, etc. [1]. IRC users with malicious behaviors can easily hide their IP address when connected to IRC channels. Moreover, unlike most social network service requiring a sign-up task, IRC network allows the user to connect channels via a non-registration process where a user can easily change nickname as they wish. When hackers leverage anonymity to hide their identities, network forensics may fail. In such a situation, the messages communicated in IRC channels may be the only clue to identify hackers. For instance, British police captured several cybercriminals actively involving in "Operation Payback," the DDoS campaign launched by the Anonymous organization in IRC by analysis of the messages of users in AnonOps channel [20]. Hence, it is critical import to develop effective social media forensic methods which can monitor, analyze, and identify hackers in IRC.

The most common social media forensics framework for identifying users is a closed-set author attribution problem that can be defined as attributing text samples of unknown authorship to one of the suspects when given a set of clear candidate authors [16]. This framework is suitable for social media forensic cases when a specific set of candidates can be provided according to certain restrictions such as prior knowledge of specific fact and access specific material. Another demanding issue is author verification which can be formulated as follows: there is only one candidate author who has a set of text samples of known authorship and the task is to verify if an unknown text sample is written by that particular author [17], [26]. One-class classification approaches try to identify objects of a specific class amongst all objects by learning from training samples containing only the objects of that class [15], [26], [27]. Therefore, author verification can be treated as a one-class classification problem where the text samples written by candidate author represents normal samples [26], [27]. While all text samples from other authors represent abnormal samples [26], [27]. Although there are some author attribution studies in IRC, the research of author verification in IRC channels is very limited. Hence, in order to fill this research gap, we propose a novel author verification approach to use one-class classification with deep autoencoder. In addition, we also

978-1-7281-5052-9/19/$31.00 ©2019 IEEE

perform recursive deep learning for classifying IRC chat message into different threat levels (i.e., normal, warning, and high) for determining dangerous users.

The remainder of this paper is structured as follows. In Section II we provide background information about IRC client, author verification, and Watson Platform. Section III explains our proposed architecture. The experimental setups and performance results are presented in Section IV. Finally, in Section V, we conclude the paper.

II. Background and Related Work

A. Internet Relay Chat (IRC)

Internet Relay Chat (IRC) is a popular communication method, especially in the cyber domain. IRC requires a server that provides networking for connected users through a protocol that facilitates real-time text communications [1]. IRC has been traditionally utilized for legitimate functions, but it has also been extensively used by hacker, anonymity, and terrorist over the years [1], [25]. IRC provides two methods of communication: (i) private chat and (ii) broadcasted public messages. In the channels, public messages sent by the users are broadcasted to all other users in the same channel in real-time. Hence, this differs from the website behavior because on the websites (e.g., blogs), the users can read previously posted messages anytime by browsing them [24]. On the contrary to the website blogs where offline collection and batch processing would work efficiently, in IRC based communication, real-time collection and threat detection are critical research issues [1], [23], [38].

In this research, in order to monitor the IRC channels, we have developed autonomic IRC monitoring bots for the comprehensive real-time collection and classification of the IRC messages using several strategies as will be discussed in Section III.

B. Author Verification

Author verification is a subarea of author identification that has been widely used for various reasons such as computer forensics, plagiarism check, social media misuse, etc. Author identification of online messages is a particularly important issue in cybercrime because one of the obvious features of cybercrime is anonymity. Anonymous users always fake their personal information and hide their identity for escaping from security investigation.

Most of the author identification works focus on author attribution problem. For example, Zheng et al. [31] developed an author identification framework based on writing-style features from lexical features, syntactic features, word-based features, structural features, and content-specific features. They performed experiments up to 20 of most active users who frequently posted messages in online newsgroups forum. Abbasi et al. [32] provided an approach called Writeprints based on the extension framework of reference [31]. Instead of using the same features for all authors, they created individual feature sets for each according to the individual's key stylometric features.

Previous author identification problem has been studied extensively in author attribution problem. However, limited work has been studied on author verification, especially for IRC. Stamatatos et al. [3] first discussed the author verification problem. According to a dataset of newspaper articles, they applied a regression method to verify authors. Koppel et al. proposed an author verification method called Unmasking which treats author verification as a one-class classification problem [26], [27]. They use support vector machine (SVM) classifier to distinguish the text with unknown authorship from a set known text. Then, they remove the most important feature and repeat this procedure. The unknown text and known text are determined by the same author when the accuracy of SVM significantly drops. Luyckx et al. approximated the author verification as a binary classification problem where they consider available texts by other authors as training samples of abnormal class [28]. Escalante et al. used particle swarm model selection for selecting an ad-hoc classifier for verifying each author in an automatic way [29]. Brocardo et al. extracted stylometry features and then trained a support vector machine model for verifying email and microblog users [34]. Barbon et al. applied the k-nearest neighbors model for performing author verification on Twitter [33]. Litvak used convolutional neural network model for author verification on an email dataset [35]. Boenninghoff et al. proposed Siamese networks topology for similarity learning on the author verification task [30].

C. Watson Platform

Watson is the artificial intelligence (AI) platform service provided by IBM to allow users to integrate AI into their applications, training, management, and analysis of data in a secure cloud environment (guaranteeing the privacy of the data) [2], [4]. We leveraged the IBM Watson Assistant and Personality Insight capabilities to build the conversation module and personality feature extraction module for our autonomic IRC monitoring bot and author verification learning unit. Watson Assistant is an AI assistant service for social media to answer questions through pre-configured content intents (e.g., banking) [2]. Furthermore, the service can also be improved using interactions history [2]. Another service we leveraged in our approach is the IBM Personality Insights that is based on integrating psychology and data analytics algorithms to analyze the given content and create a personality profile [4]. The IBM Personality Insights service uses three models: Big Five, Needs, and Values [4]. Big Five personality characteristics represent the most widely used model for generally describing how a person engages and interacts with the world. This model includes five primary dimensions based on [4] as follows. (1) Agreeableness: a person's tendency to be compassionate and cooperative toward others; (2) Conscientiousness: a person's tendency to act in an organized or thoughtful way; (3) Extraversion: a person's tendency to seek stimulation in the company of others; (4) Emotional range, also referred to as Neuroticism or Natural reactions: the extent to which a person's emotions are sensitive to the person's environment; and (5) Openness: the extent to which a person is open to experiencing a variety of activities. Each of these top-level dimensions has six facets that further characterize an individual according to the dimension. Needs model describes which aspects of a product will resonate with a person and includes twelve characteristic needs: Excitement, Harmony, Curiosity, Ideal, Closeness, Self-expression, Liberty, Love, Practicality, Stability, Challenge, and Structure [4]. Values model describes motivating factors that influence a person's decision-making process. The model includes five values: Self-transcendence, Conservation, Hedonism, Self-enhancement, Open to change [4]. Watson infers personality features from textual information using an open-vocabulary approach. By using GloVe, which is an open-source word embedding technique, the service obtains a vector representation for the words in the input text. It then feeds this representation to a machine learning model that infers a personality profile. To train the model, IBM uses scores from surveys that were conducted among thousands of users along with their Twitter data [4].

III. An Architecture of Author Verification in IRC

In this section, we present the framework design including autonomic IRC monitoring bot, feature extraction for author verification, and learning unit. The architecture of our framework is shown in Figure 1.

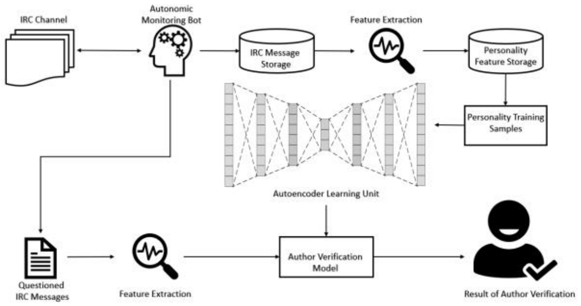
IRC Channel
Autonomic Monitoring Bot
IRC Message Storage
Feature Extraction
Personality Feature Storage
Personality Training Samples
Autoencoder Learning Unit
Questioned IRC Messages
Feature Extraction
Author Verification Model
Result of Author Verification

Figure 1. The architecture of author verification in IRC

A. Autonomic IRC Monitoring Bot

As IRC is a real-time communication method in the form of text, the monitoring should be in real-time [1], [36]. In order to tackle the collection and classification issue, we developed the autonomic IRC bot as shown in Figure 2. The autonomic IRC bot has capabilities of robust continuous monitoring, comprehensive information collection, pre-processing, and threat level classification in real-time. With these capabilities, the bot monitors the channel and transforms the unstructured IRC data to structured data with the following CSV format: ***Threat level*** + ***User nickname*** + ***Chat content*** + ***Date*** + ***Time***.

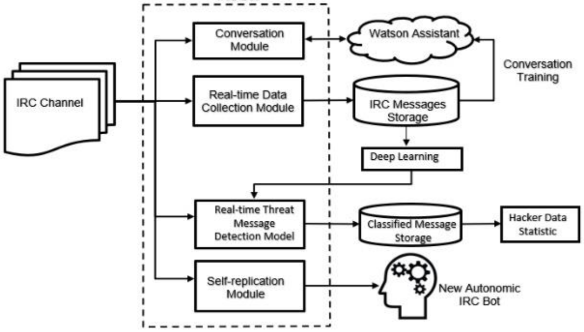

Figure 2. The architecture of autonomic IRC monitoring bot

1) Real-time Message Collection

We created a conversation module providing basic response capabilities so that the bot can escape the identification by IRC channel operators since the operators sometimes ask questions and expect valid answers in short amount of time (and if not responded, the operators can block). With the IBM Watson Assistant service [2], we built a machine learning model that understands the chat input and respond to IRC users in a way that simulates conversations between humans. In addition, many hacker channels publish self-signed certificate and enforce the strict standard of using TLS/SSL to access to their IRC network. To overcome this issue, the functions that can trust all the self-signed certificate and specify the port number for SSL/TLS connections are added to the IRC bot. It is also possible that the cybercriminals create a temporary channel where they can diffuse illegal content or propagate hacking tools, and even launch organized cyberattacks such as Denial of Service (DoS). Therefore, to continuously monitor such threat activities, a self-replication module is developed, allowing parent bot to generate a new (child) bot that inherits all the capability when detecting a channel name in IRC message.

2) Deep Learning for Real-time Message Classification

The Recursive Neural Tensor Network (RNTN) has been proven to be powerful for sentiment analysis task in [6]. RNTN deep learning model represents words and phrases as D-dimensional vectors through recursively performing tensor-based composition function to the parse tree. Words of a sentence are represented as numeric vectors and combined by tensor-based composition function to form parent vectors in a bottom-up way. The vector of node i is calculated by:

$$V^i = f\left(\begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix}^T T^{[1:D]} \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix} + W \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix} + b\right) \qquad (1)$$

where V_l^i, V_r^i are the vectors of current node's left child node and right child node. $W \in \mathbb{R}^{D\times 2D}$ is a linear composition matrix, $T^{[1:D]}$ is a tensor, b is the bias vector, and f is a non-linear activation function. The tensor product is given by:

$$\begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix}^T T^{[1:D]} \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix}^T T^{[1]} \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix} \\ \vdots \\ \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix}^T T^{[D]} \begin{bmatrix} V_l^i \\ V_r^i \end{bmatrix} \end{bmatrix} \qquad (2)$$

where $T^{[d]} \in \mathbb{R}^{2D\times 2D}$ is a slice of tensor $T^{[1:D]}$. The vectors and tensor-based composition function are updated through backpropagation algorithm. The softmax function is used as the final layer to normalize an input vector into a probability distribution of k class.

The CoreNLP library [7] is used to achieve RNTN deep learning model. After integrating RNTN model, the autonomic IRC bot can automatically distinguish between normal chat messages and threat chat messages, and further, identify the threat level of the chat messages. In order to train the recursive deep learning model, we use messages collected from IRC channels. We classify each IRC message into three levels: Normal, Warning, and High with each level is given the sores of 0, 1, 2, respectively. Normal (Score 0) represents messages being interactive with other contacts without having any malicious content. Warning level (Score 1) indicates potential risk that has been detected because of the use of hacking terms. High level (Score 2) denotes that the sender appears to have some malicious behaviors or intentions to perform malicious activities. The labeling rule for the current node score depends on the threat level of the whole phrase that current node dominates under the parse of the threat tree. For example, in Figure 3, the representation of "library network" is calculated by the tensor-based composition function of "library" and "network", and the representation of "of library network" is recursively calculated through the vectors of "of" and "library network".

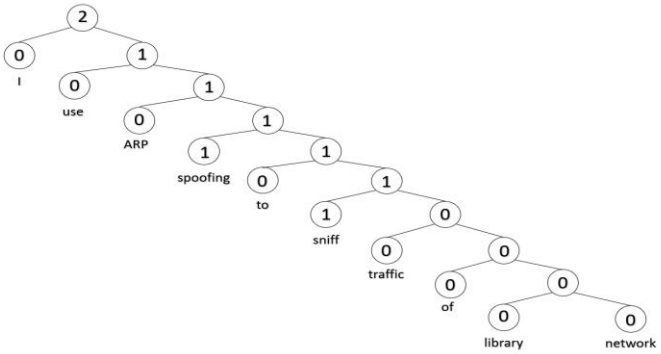

Figure 3. An example of our labeling rule, from normal to high (0, 1, 2) at every node of the threat tree construction.

The steps of performing recursive deep learning-based threat classification are listed as follows: (1) Tokenize the IRC message into a sequence of words which are represented as numeric vectors; (2) Generates the lemmas (base forms) for words; (3) Tag words and phrases with part of speech tagger (POS); (4) Parse the IRC message into its constituent phrases and words and build a syntactic threat tree; (5) Classify the IRC message threat level using the recursive neural tensor network. All the nodes of the IRC message, especially, the root node, are given a threat level score. A test set which contained 906 IRC messages was created. The results of recall and precision of each threat level in our test set are reported in Table I.

Table I. MODEL PERFORMANCE EVALUATION ON TEST SET

	Normal (root node)	Warning (root node)	High (root node)
Percentage in test set	74.83%	19.65%	5.52%
Recall	89.09%	84.83%	76.00%
Precision	98.37%	70.23%	49.35%

B. Feature Extraction

The messages can be measured to characterize the individual user's personality. The personality characteristics are distinguished uniquely from individual to individual and relatively stable [5]. Based on how IRC users communicate with others, personality characteristics influence most of the user's activities and behaviors in the IRC channel. Moreover, personality also influences the way IRC users make decisions including cyberattack types and hacking techniques selection, attack and crime motivation, hacking organizations, and malicious tools being developed, which are expressed in their messages [25]. Using IBM's Personality Insights services as explained in Section II, we can successfully analyze individual authors' IRC messages and intrinsic personality characteristics to create their personality profiles. Our personality feature extractor can operate using different languages as IBM Personality Insights service supports multiple languages (e.g., English, Japanese, Korean, Arabic), which is important for international cybersecurity investigations. In this work, we have used English only (the other languages should be straight-forward). By calling the Personality Insights service from IBM Cloud, the personality analysis module can get an individual user's personality insights in JSON format (that has the normalized personality analysis results based on three models: Big Five, Needs, and Values). Big Five model contains the following five primary dimensions: Agreeableness, Conscientiousness, Extraversion, Emotional range, and Openness. Each of these primary dimensions includes six facet features to further distinguish a user. Needs model contains twelve need features, and Values model includes five value features. We selected all the facet features and primary features of Big Five, all the features of Needs model, and all the features of Values model to represent the personality of the suspect user, which creates 52 features in total. Watson recommends that user provides 1,200 words for personality analysis, but providing 600 words produces acceptable results and 3.000 words are sufficient to achieve the maximum precision [4]. A sunburst chart visualization for a user's personality profile is shown in Figure 4.

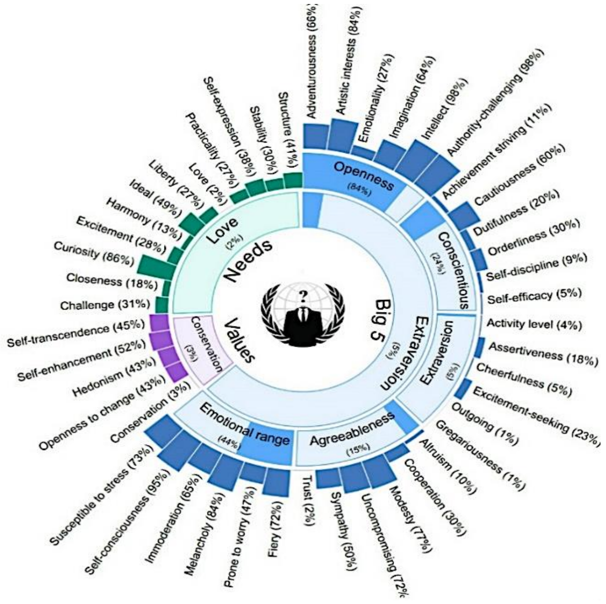

Figure 4. Visualization of an IRC user's personality insights features

C. Learning Unit

To compare the performance of one-class classification with deep autoencoder and other models, we have adopted two important outlier detection models, which are one-class support vector machine and isolation forest.

1) One-class Classification with Deep Autoencoder

An autoencoder is a feedforward network for learning a map from the input to itself through a pair of encoding and decoding phases [11]. A hidden layer in an autoencoder neural network generates a bottleneck enforcing autoencoder to compress input to a low-dimensional representation. For each input sample, we aim to train the autoencoder to make output and input as closely as possible. Autoencoder can be trained using the backpropagation algorithm with the output value to be equal to the input value. The forward propagation of a simple autoencoder with one hidden layer is given by:

$$h = f\left(W^{(1)}x + b_1\right) \qquad (3)$$

$$y = f\left(W^{(2)}h + b_2\right) \qquad (4)$$

where h is the vector of representation of the hidden node activities and f is a non-linear activation function. $W^{(1)}$ is a parameter matrix between the input layer and hidden layer and b_1 is a vector of bias parameters. $W^{(2)}$ is a parameter matrix between the hidden layer and output layer and b_2 is a vector of bias parameters. x is a vector representing input and y is a vector representing the output. Autoencoder learns the parameters by performing gradient descent to minimize the reconstruction error. The squared sum of the error in reconstruction of the sample is used to measure the reconstruction error in our approach.

An autoencoder with multiple hidden layers can be called a deep autoencoder [11], [12]. Deep autoencoder uses multiple hidden layers to perform non-linear transformation for generating multiple representation spaces. Therefore, deep autoencoder can effectively represent complicated distribution over the input [11]. The deep autoencoder is also a feedforward artificial neural network. Therefore, we can train deep autoencoder through the backpropagation algorithm [11].

The architecture of our deep autoencoder with symmetrical shape is shown in Figure 5. The input layer has 52 nodes since we extract 52 personality insights features. Therefore, the output layer also has 52 nodes since the purpose is to reconstruct sample. Five fully connected layers (whose nodes are 30, 20, 10, 20, and 30, respectively) are added into autoencoder as hidden layers for forming a deep autoencoder. The middle hidden layer with the 10 nodes as the code layer which stores the compressed representation space for the input data. Our deep autoencoder uses tanh activation function for each hidden layer and identity function for the output layer. The deep autoencoder learns the parameters by using Adam optimizer with mini-batch training to minimize the mean squared error. After completing the training stage, our deep autoencoder model can verify the authorship of IRC message. Whether an IRC message sample belongs to a user is determined by reconstruction error. In the test stage, an IRC message sample has low reconstruction error if it is written by the same author. While the reconstruction error is large if it is written by a different author.

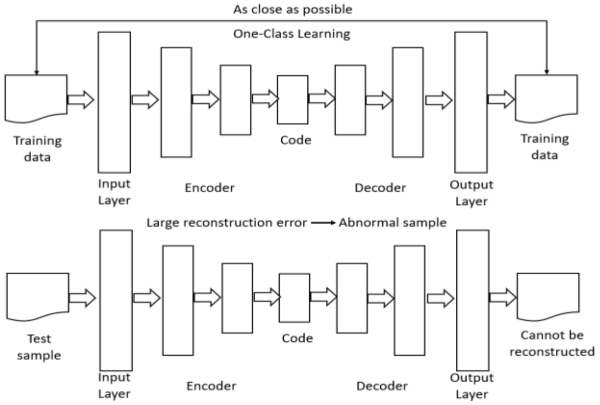

Figure 5. The architecture of deep autoencoder for author verification

2) One-class Support Vector Machine

One-class support vector machine (OC-SVM) is a one-class classification algorithm developed by Schölkopf et al. [9]. This algorithm map data into a feature space using a kernel function and try to separate the mapped features from the original space with maximum margin. In this paper, we use the Radial Basis Function (RBF) kernel because the number of input features is 52 which is not large. RBF is given as:

$$K\left(x_i, x_j\right) = exp\left(-\gamma\left\|x_i - x_j\right\|^2\right) \qquad (5)$$

Given the training data $x_i \in R^n, i \in [m]$, the algorithm can be obtained as follows:

$$\min_{w,\xi,\rho} \frac{1}{2}w^T w - \rho + \frac{1}{\upsilon m}\sum_{i=1}^{m}\xi_i \qquad (6)$$

$$\text{Subject to } w^T\phi(x_i) \geq \rho - \xi_i, \xi_i \geq 0,$$

where ρ represents the margin, w is a vector orthogonal to the hyperplane, and ξ_i are slack variables. υ represent an expected fraction of training samples that are allowed to be rejected. The decision function is given by:

$$sign\left(\sum_{i=1}^{m}\alpha_i K(x_i, x) - \rho\right) \qquad (7)$$

where the α_i is coefficients. The training samples x_i for which $\alpha_i \neq 0$ are treated as support vector.

3) Isolation Forest

Isolation Forest (IF) is an anomaly detection model for isolating abnormal samples [10]. IF can build an ensemble of isolation trees. During the iteration of building every tree, IF randomly select a feature from the random subset and then selecting a random split value between the maximum value and minimum value of this feature. The splitting ends up with the condition that every tree node containing only one sample. The path length is the number of splitting needed to isolate the sample. Outliers are more susceptible to isolation and therefore have a small number of path length. Hence, path length can be used to represent a measure of abnormality [10]. IF can use subsampling to achieve a low linear time-complexity and a small memory-requirement. IF also works well in high dimensional data with a large number of irrelevant features and when abnormal samples are not available in training set [10].

IV. Experiments and Results

To evaluate our approach, we performed author verification tasks on 12 highly heterogeneous IRC channels that are monitored by our autonomic IRC monitoring bots (shown in Table II). The monitored channels are as follows:

- The #anonops channel is an international communication platform controlled by the infamous anonymous hacking organization which is involved in many cybercrimes.
- The #2600 channel is a highly active community with hacker magazines and monthly hacker meetings.
- The #darkscience and #hak5 are two main hacker channels, which also provide tor hidden connection, in the server Darkscience that is a hacker community sharing the interest in security.
- The #computer is the main discussion channel for the Underworld server for understanding cybercrimes and immoral deeds on the Internet.
- The #thepiratebay.org is a channel that focuses on the topic of copyright infringement and participating in the swarm for illegal files.
- The #trading channel is another Underworld's important channel where the topics appear in the black markets.
- The #security, #networking, and #bash are three popular channels involving the topics of cybersecurity in the Freenode server.
- The #bitcoin is the channel that appears topics on using anonymous currency in the black markets and illegal tradings.
- The #undernet is the channel that discusses the topic related to hacking and cracking in the Undernet server.

Table II. TOTAL # OF MESSAGES OF THE MONITORED CHANNELS

Server Name	Channel Name	Total # of Messages	Collection Date Range
irc.anonops.com	#anonops	1,326,580	8/15/17 – 9/20/18
irc.2600.net	#2600	785,849	4/01/17 – 9/20/18
irc.darkscience.net	#darkscience	165,896	10/11/17 – 9/20/18
irc.darkscience.net	#hak5	128,389	4/01/17 – 9/20/18
irc.underworld.no	#computer	361,320	9/13/17 – 9/20/18
irc.underworld.no	#thepiratebay.org	81,095	3/13/18 – 9/20/18
irc.underworld.no	#trading	75,996	11/28/17 – 9/20/18
irc.freenode.net	#networking	338,493	9/06/17 – 9/14/18
irc.freenode.net	#security	289,007	4/13/17 – 9/20/18
irc.freenode.net	#bash	266,098	9/06/17 – 9/20/18
irc.freenode.net	#bitcoin	164,663	9/13/17 – 9/20/18
irc.undernet.net	#undernet	310,723	12/05/17 – 9/20/18

From these 12 cybersecurity-related IRC channels, which are collected and classified in real-time, we extract the different users with more than 10,000 words for experiments. In this way, 441 different users are extracted for author verification in IRC. Then each user's whole IRC messages are performed by non-overlap segment for generating multiple samples. After segmentation, 19,004 samples are generated where each sample has equal 1000 words (we discarded the remaining messages that are less than 1,000 words). Then, we analyze the personality features of each sample through personality insights.

We ranked the potential malicious users according to the summation of the threat level score (normal=0, warning=1, and high=2) of all their own IRC messages. Then, the top 30 potential malicious users are selected as the target author need to be verified. Hence, the experimental dataset has 441 different class with 19,004 samples where we create 30 one-class classification-based author verification experiments. In each experiment, one of the 30 target authors is the normal class and samples from the remaining 440 classes are represented as an abnormal class. Each experiment is repeated 10 times using 80/20 training and test split according to 10 random states. For the training set, we only train the samples from the respective normal class. IF, one-class support vector machine (OC-SVM), and deep autoencoder (DAE) are used to create author verification models in each experiment. Feature standardization is used to standardize the range of personality insights features. The formula is given as:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (8)$$

where x is the original feature vector, $\bar{x}$ is the mean of that feature vector, and σ is the standard deviation of that feature vector.

The area under the curve (AUC) and average precision (AP) are used to measure the performance of author verification models. AUC summarizes the entire location of the receiver operating characteristic (ROC) curve which shows the tradeoff between true positive rate and false positive rate for different thresholds. AP summarizes the precision-recall curve which shows the tradeoff between precision and recall curve for different thresholds [14]. Results are shown in Table III. We notice that DAE outperforms IF and OC-SVM in both AUC and AP measurements. For the AUC measurement, we observe that DAE uniquely achieved the highest AUC in 27 users. As the remaining users, there are two users where OC-SVM and DAE achieved the equal best AUC. And there is one user where OC-SVM slightly better than DAE. For the AP measurements, DAE uniquely approached the highest AP in 24 users. Besides, there are five users where OC-SVM and DAE achieved the equal best AP. As the remaining one user, OC-SVM slightly better than DAE. We also notice that: although DAE is generally considered a powerful deep learning model on large training samples, our approach still perform well for the user who does not have a large number of training samples, such as the user whose total number of samples in the dataset is less than 100.

TABLE III AVERAGE AUC AND MEAN AP PER METHOD OVER TEN RANDOM STATES; SUMMATION OF THREAT LEVEL SCORE PER USER; TOTAL # OF SAMPLES PER USER IN DATASET

			AUC			AP		
Normal Class (Nickname)	Sum of threat level score	Total # of samples in dataset	IF	OC-SVM	DAE	IF	OC-SVM	DAE
JARVIS	13126	370	99.3484%	99.5406%	**99.7227%**	99.9863%	99.9908%	**99.9946%**
Meow	12524	212	99.4511%	99.5955%	**99.6978%**	99.9928%	99.9948%	**99.9961%**
Gopher	5006	592	88.8388%	88.4675%	**93.6629%**	99.5223%	99.5269%	**99.7586%**
thufir	4908	62	99.9892%	**100.0000%**	**100.0000%**	99.9999%	**100.0000%**	**100.0000%**
aestetix	4213	513	94.2906%	94.4549%	**96.2867%**	99.8101%	99.8208%	**99.8808%**
Effexor	3703	610	91.8959%	93.2563%	**95.5548%**	99.8763%	99.9004%	**99.9239%**
zeta	3411	81	99.2788%	99.7391%	**99.7804%**	99.9963%	**99.9986%**	**99.9986%**
piqure	3228	38	97.6161%	99.4444%	**100.0000%**	99.9762%	99.9974%	**100.0000%**
WolfBot	3183	44	97.4538%	99.2113%	**99.4755%**	99.9934%	99.9980%	**99.9987%**
RDNt	3150	228	80.2582%	80.5427%	**84.0504%**	99.6397%	99.6460%	**99.7155%**
LostBiT	2973	128	99.7429%	**100.0000%**	**100.0000%**	99.9969%	**100.0000%**	**100.0000%**
catphish	2700	173	92.3049%	92.5589%	**94.2862%**	99.9109%	99.9229%	**99.9417%**
keiththewhteguy	2587	253	96.5315%	96.9718%	**97.2634%**	99.9425%	99.9515%	**99.9577%**
druqs	2493	69	99.5174%	99.9452%	**99.9835%**	99.9978%	99.9998%	**99.9999%**
stovepipe	2451	374	87.9864%	88.2338%	**92.0199%**	99.6730%	99.6686%	**99.7604%**
shbot	2207	38	99.9188%	99.9993%	**99.9996%**	99.9998%	**99.9999%**	**99.9999%**
covfefe	2047	245	92.3746%	92.3838%	**93.2877%**	99.8813%	99.8781%	**99.8923%**
djph	1997	135	96.1739%	96.6302%	**96.9671%**	99.9666%	99.9713%	**99.9741%**
greycat	1869	166	96.0876%	96.6245%	**97.5139%**	99.9620%	99.9680%	**99.9766%**
maestro	1822	230	83.2116%	84.6795%	**88.2934%**	99.7152%	99.7471%	**99.8117%**
twelve	1791	188	95.1658%	96.3289%	**98.5323%**	99.9429%	99.9591%	**99.9844%**
greybot	1775	192	99.9590%	99.9858%	**99.9895%**	99.9996%	**99.9999%**	**99.9999%**
WildMan	1754	63	96.6923%	98.7102%	**99.5778%**	99.9794%	99.9948%	**99.9986%**
Cogitabundus	1738	222	94.4273%	94.5253%	**96.8387%**	99.9178%	99.9118%	**99.9447%**
Crono_	1724	402	94.3800%	95.2420%	**97.3852%**	99.8460%	99.8858%	**99.9386%**
Cochise	1678	147	98.0231%	**98.4855%**	98.2707%	99.9717%	**99.9807%**	99.9774%
dmt	1589	207	95.5832%	95.7013%	**96.2176%**	99.9464%	99.9497%	**99.9550%**
catface	1572	165	83.8690%	86.0893%	**88.3075%**	99.7985%	99.8324%	**99.8611%**
mickers	1544	178	84.6574%	84.8641%	**88.5190%**	99.7908%	99.8031%	**99.8588%**
lazarus	1515	74	95.9817%	98.2243%	**98.7986%**	99.9816%	99.9919%	**99.9944%**

V. Conclusion

The explosive growth of IT infrastructures, cloud systems, mobile devices, and Internet service have resulted in cyber-threats that are growing exponentially in the number and also in the complexity nowadays [8], [18], [19]. A growing number of cybercriminals are leveraging anonymous techniques such as masking IP address and communicating through Tor [39]. These techniques are designed to hamper the investigative effort. Authorship analysis is a promising approach to identify cybercriminals who use anonymous communication. For example, author verification can verify the relationships between observed malicious data and attackers and therefore provide the capability to assist the Dynamic Data-Driven Application System (DDDAS) for cyber trust analysis [13], [21], [40], [41]. IRC has been one of the important platforms for rising cyber-threats by performing threat communications. As a result, it is highly desired to design method that can verify the authorship between threat communications and human actors. In this paper, we first collect and classify IRC messages using our autonomic IRC monitoring bot. Then we develop a novel author verification approach via one-class learning with deep autoencoder. We show the effectiveness of our author verification approach on dangerous users in IRC cybersecurity-related channels.

Acknowledgment

This work is partly supported by the Air Force Office of Scientific Research (AFOSR) Dynamic Data-Driven Application Systems (DDDAS) award number FA9550-18-1-0427, National Science Foundation (NSF) research projects NSF-1624668 and NSF-1849113, (NSF) DUE-1303362 (Scholarship-for-Service), National Institute of Standards and Technology (NIST) 70NANB18H263, and Department of Energy/National Nuclear Security Administration under Award Number(s) DE-NA0003946.

References

[1] V. Benjamin, W. Li, T. Holt, and H. Chen, "Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops," in 2015 IEEE Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on, 2015, pp. 85-90.

[2] "IBM Watson Assistant service," [Online] URL: https://www.ibm.com/watson/services/conversation/, Accessed: December 2017

[3] E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Automatic text categorization in terms of genre and author," Computational linguistics 26, no. 4 (2000): 471-495.

[4] "IBM Watson Personality Insights service," [Online] URL: https://console.bluemix.net/docs/services/personality-insights, Accessed: December 2017

[5] D. A. Cobb-Clark, and S. Schurer, "The stability of big-five personality traits," *Economics Letters*, vol. 115, no. 1, 2012.

[6] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," In Proceedings of the conference on empirical methods in natural language processing, pp. 1631-1642. 2013.

[7] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55-60. 2014.

[8] C. Tunc, S. Hariri, F. D. L. P. Montero, F. Fargo, and P. Satam. "CLaaS: Cybersecurity Lab as a Service--Design, Analysis, and Evaluation," In 2015 International Conference on Cloud and Autonomic Computing, pp. 224-227. IEEE, 2015.

[9] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. "Estimating the support of a high-dimensional distribution," Neural computation 13, no. 7 (2001): 1443-1471.

[10] F. T. Liu, K. Ting, and Z. Zhou. "Isolation-based anomaly detection." ACM Transactions on Knowledge Discovery from Data (TKDD) 6, no. 1 (2012): 3.

[11] C. Zhou, and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665-674. ACM, 2017.

[12] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning," nature 521, no. 7553 (2015): 436.

[13] Y. Badr, S. Hariri, Y. AL-Nashif, and E. Blasch, "Resilient and trustworthy dynamic data-driven application systems (DDDAS) services for crisis management environments," Procedia Computer Science 51 (2015): 2623-2637.

[14] "Sklearn.metrics.precision_recall_curve," [Online] URL: https://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html, Accessed: April 2019

[15] S. Khan, and M. G. Madden. "A survey of recent trends in one class classification," In Irish conference on artificial intelligence and cognitive science, pp. 188-197. Springer, Berlin, Heidelberg, 2009.

[16] E. Stamatatos, "Author identification: Using text sampling to handle the class imbalance problem." Information Processing & Management 44, no. 2, 2008

[17] H. V. Halteren, "Linguistic profiling for author recognition and verification," In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 199. Association for Computational Linguistics, 2004.

[18] C. Tunc, and S. Hariri. "CLaaS: Cybersecurity Lab as a Service." J. Internet Serv. Inf. Secur. 5, no. 4 (2015): 41-59.

[19] C. Tunc, S. Hariri, F. D. L. P. Montero, F. Fargo, P. Satam, and Y. Al-Nashif, "Teaching and Training Cybersecurity as a Cloud Service," In 2015 International Conference on Cloud and Autonomic Computing, pp. 302-308. IEEE, 2015.

[20] "UK cops: How we sniffed out convicted AnonOps admin 'Nerdo,'" [Online] URL: https://www.theregister.co.uk/2012/12/14/uk_anon_investigation/ Accessed: October 2018

[21] S. Hariri, C. Tunc, P. Satam, F. Al-Moualem, and E. Blasch, "DDDAS-Based Resilient Cyber Battle Management Services (D-RCBMS)," In Proceedings of the 2015 IEEE 22nd International Conference on High Performance Computing Workshops (HiPCW), pp. 65-65. IEEE Computer Society, 2015.

[22] J. Martin, "Lost on the Silk Road: Online drug distribution and the 'cryptomarket'," *Criminology & Criminal Justice.,* vol. 14, no. 3, pp. 351-367, 2014.

[23] S. Shao, "Real-Time Automatic Framework for IRC Threat Information Detection," In 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W), pp. 382-384. IEEE, 2017.

[24] V. Benjamin, B. Zhang, J. F. Nunamaker Jr, and H. Chen, "Examining hacker participation length in cybercriminal Internet-relay-chat communities," Journal of Management Information Systems 33, no. 2 (2016): 482-510.

[25] S. Shao, C. Tunc, A. Al-Shawi, and S. Hariri, "Autonomic Author Identification in Internet Relay Chat (IRC)," In 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1-8. IEEE, 2018.

[26] M. Koppel, and J. Schler. "Authorship verification as a one-class classification problem," In Proceedings of the twenty-first international conference on Machine learning, p. 62. ACM, 2004.

[27] M. Koppel, J. Schler, and E. Bonchek-Dokow. "Measuring differentiability: Unmasking pseudonymous authors." Journal of Machine Learning Research 8, no. Jun (2007): 1261-1276.

[28] K. Luyckx, and W. Daelemans, "Authorship attribution and verification with many authors and limited data," In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 513-520. Association for Computational Linguistics, 2008.

[29] H. J. Escalante, M. Montes, and L. Villaseñor. "Particle swarm model selection for authorship verification." In Iberoamerican Congress on Pattern Recognition, pp. 563-570. Springer, Berlin, Heidelberg, 2009.

[30] B. Boenninghoff, R. M. Nickel, S. Zeiler, and D. Kolossa, "Similarity Learning for Authorship Verification in Social Media," In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2457-2461. IEEE, 2019.

[31] R. Zheng, J. Li, H. Chen, and Z. Huang. "A framework for authorship identification of online messages: Writing ‐ style features and classification techniques," Journal of the American society for information science and technology, vol. 57, no. 3, 2006.

[32] A. Abbasi, and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," ACM Transactions on Information Systems, vol. 26, no. 2, 2008.

[33] S. Barbon, R. A. Igawa, and B. B. Zarpelão, "Authorship verification applied to detection of compromised accounts on online social networks," Multimedia Tools and Applications 76, no. 3 (2017)

[34] M. Brocardo, I. Traore, and I. Woungang, "Authorship verification of e-mail and tweet messages applied for continuous authentication," Journal of Computer and System Sciences 81, no. 8 (2015): 1429-1440.

[35] M. Litvak, "Deep Dive into Authorship Verification of Email Messages with Convolutional Neural Network," In Annual International Symposium on Information Management and Big Data, pp. 129-136. Springer, Cham, 2018.

[36] J. Bernard, S. Shao, C. Tunc, H. Kheddouci, and S. Hariri, "Quasi-cliques Analysis for IRC Channel Thread Detection," In International Conference on Complex Networks and their Applications, pp. 578-589. Springer, Cham, 2018.

[37] S. Shao, C. Tunc, P. Satam, and S. Hariri. "Real-time irc threat detection framework," In 2017 IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS* W), pp. 318-323. IEEE, 2017.

[38] J. Yu, C. Tunc, and S. Hariri. "Automated Framework for Scalable Collection and Intelligent Analytics of Hacker IRC Information," In 2016 International Conference on Cloud and Autonomic Computing (ICCAC), pp. 33-39. IEEE, 2016.

[39] W. Yu, X. Fu, E. Blasch, K. Pham, D. Shen, G. Chen, and C. Lu, "On effectiveness of hopping-based spread spectrum techniques for network forensic traceback," In ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 101-106. IEEE, 2013.

[40] E. Blasch, Y. Al-Nashif, and S. Hariri, "Static versus dynamic data information fusion analysis using DDDAS for cyber security trus," Procedia Computer Science 29 (2014): 1299-1313.

[41] E. Blasch, "DDDAS advantages from high-dimensional simulation," In Winter Simulation Conference (WSC), pp. 1418-1429. IEEE, 2018.