Transfer Learning Approach for Botnet Detection Based on Recurrent Variational Autoencoder

Jeeyung Kim
Lawrence Berkeley Nat'l Laboratory
Berkeley, CA, USA
jeeyungkim@lbl.gov

Alex Sim
Lawrence Berkeley Nat'l Laboratory
Berkeley, CA, USA
asim@lbl.gov

Jinoh Kim
Texas A&M University
Commerce, TX, USA
jinoh.kim@tamuc.edu

Kesheng Wu
Lawrence Berkeley Nat'l Laboratory
Berkeley, CA, USA
kwu@lbl.gov

Jaegyoon Hahm
KISTI
Daejeon, South Korea
jaehahm@kisti.re.kr

ABSTRACT

Machine Learning (ML) methods have been widely used in Intrusion Detection Systems (IDS). In particular, many botnet detection methods are based on ML. However, due to the fast-evolving nature of network security threats, it is necessary to frequently retrain the ML tools with up-to-date data, especially because data labeling takes a long time and requires a lot of effort, making it difficult to generate training data. We propose transfer learning as a more effective approach for botnet detection, as it can learn from well curated source data and transfer the knowledge to a target problem domain not seen before. We devise an approach that is effective regardless whether or not the data from the target domain is labeled. More specifically, we train a neural network with the Recurrrent Variation Autoencoder (RVAE) structure on the source data, and use RVAE to compute anomaly scores for data records from the target domain. In an evaluation of this transfer learning framework, we use CTU-13 dataset as a source domain and a fresh set of network monitoring data as a target domain. Tests show that the proposed transfer learning method is able to detect botnets better than semi-supervised learning method that was trained on the target domain data. The area under Receiver Operating Characteristic is 0.810 for transfer learning, and 0.779 for directly using RVAE on the target domain data.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; • **Computing methodologies → Machine learning**; **Unsupervised learning**; **Anomaly detection**; **Transfer learning**; **Neural networks**.

KEYWORDS

botnet detection; transfer learning; Recurrent Neural Network; Variational Autoencoder

SNTA '20, June 23, 2020, Stockholm, Sweden

ACM ISBN 978-1-4503-7980-9/20/06.
https://doi.org/10.1145/3391812.3396273

ACM Reference Format:
Jeeyung Kim, Alex Sim, Jinoh Kim, Kesheng Wu, and Jaegyoon Hahm. 2020. Transfer Learning Approach for Botnet Detection Based on Recurrent Variational Autoencoder. In *3rd International Workshop on Systems and Network Telemetry and Analytics (SNTA '20), June 23, 2020, Stockholm, Sweden.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3391812.3396273

1 INTRODUCTION

Botnet, whose controllers hijack other devices and command a variety of cyberattacks, has become one of the most important threats to cyber security. Botnet spreads malware and ransomware by conducting distributed denial-of-service attacks (DDos), click-fraud, spamming and crypto-mining. Furthermore, the malicious software for infecting a machine and operating botnets has evolved to evade detection, resulting in various attack strategies for each botnet. Mainly, the protocols used by botnet as communication channels are different: Internet Relay Chat (IRC), peer-to-peer (P2P) and HTTP [35]. In addition, the means of distributing malware are also different from botnet to botnet. For example, a botnet of Neris relies on IRC protocols for sending spam emails or conducting click-fraud and port scanning [12]. A botnet named Virut uses HTTP protocols and has attack vectors for DDos attack and spamming. Given the severity of botnets, it is essentially required to identify such different classes of botnets to secure the cyberspace.

There have been a body of studies introduced various methodologies for botnet detection [1, 7, 13, 34], often classified into two types: signature-based and anomaly-based. A signature-based method detects malicious connections by referencing a set of rules (also known as "signatures"). Although this approach requires a relatively small amount of computation, it is significantly restricted to detect well-known botnets only [25, 34]. On the other hand, anomaly-based techniques identify botnets by detecting unusual system behaviors, such as high network latency and high volumes of traffic [34]. As a tool for anomaly detection, machine learning (ML) methods have been applied for characterizing botnet behaviors [5, 26, 29, 32, 35].

Anomaly detection based on supervised learning has shown promising results with a high degree of accuracy for detecting botnets [11, 21, 26], but one complication is that supervised learning assumes the provision of data labels to classify, which are often unavailable in practice. Indeed, anomalies in network traffic are difficult to obtain in terms of scarcity of occurrence and difficulty in classification; thus, supervised anomaly detection methods cannot

be easily adapted for botnet detection in a real world. Another line of anomaly detection based on ML, such as autoencoders (AEs) [10], Variational Autoencoder (VAEs) [3, 19, 20] and one-class support vector machines (OSVMs) [20], relies on semi-supervised learning that constructs a learning model only using normal profiles, which is much straightforward to collect. However, the detection performance is generally much lower than supervised learning techniques.

Another possible solution to address the shortage of labels would be the use of *transfer learning*, which utilizes labeled data available in another domain ("source domain") for the domain of interest ("target domain"). Even in the situation where there is insufficient labeled data in the target domain, transfer learning allows us to construct a learning model without the expensive data-labeling effort via knowledge transfer [22]. With this benefit, several transfer anomaly detection methods have been proposed in [4, 9, 14, 33]. However, the previous studies focused on text classification, speech recognition, and image classification. Few studies cope with applying transfer learning for botnet detection [2, 6, 15, 17, 27, 30]. However, these previous methods tend to depending on naive techniques, such as calculating similarity or heuristic methods, where it is hard to expect reliable performance [2, 15]. Furthermore, some of them require both normal and anomalous instances for source and target domains, which limits its utilization [15, 27].

In this paper, we propose a novel botnet detection method that can be performed on cases with no anomalies in the target domain. Moreover, we propose a training method that does not need to be marked as normal/abnormal in the target domain. In particular, we use Recurrent Variational Autoencoder (RVAE) model to obtain anomaly scores for instances from both source and target domains. By applying transfer learning framework to botnet detection, we suggest a practical methodology that is vulnerable to be utilized without efforts in labeling network traffic data.

We have two main contributions in this paper:

- We present a transfer learning framework for botnet detection that is capable to construct a learning model without the need of the label information for the associated data in the target domain.
- We verify that the presented detection approach can detect potential botnets in the new network monitoring data set as the target domain with the knowledge transferred from the popular data set of CTU-13 as the source domain. The experimental results show that the presented method detects suspicious botnet connections effectively.

2 PRELIMINARY

2.1 Transfer Learning

Transfer learning is one of machine learning techniques which are utilized in the situation where we have a classification or regression tasks in one domain of interest, but we only have sufficient labeled data in different domains, where the latter data may follow a different data distribution [22]. The transfer learning can be divided into three categories according to source/target domains label existence and the types of tasks.

Inductive transfer learning represents the case where the source/target domains are the same, but the tasks of source/target domains are different. On the other hand, *transductive transfer learning* indicates the case where the source/target domains are different but related, and the tasks of target/source domain are the same. In this case, while usually source domain labels are available, target domain labels are unavailable. *Unsupervised transfer learning* is that not only the domains of source/target domain but also the tasks of source/target domain are different.

2.2 Recurrent Variational Autoencoder

RVAE is the structure of combining seq2seq with VAE, whose encoder and decoder consist of auto-regressive model. As it utilizes RNN structure to generate outputs, it takes into account not only current inputs but also its neighborhood while generating. For prior distribution, it uses Gaussian distribution like VAE. The last hidden state is used as mean and variance of multivariate Gaussian in the latent space. The latent variable is used as the initial hidden state of the decoder. The more detailed discussion of Recurrent VAE can be referred in [8] and [24].

We use the RVAE structure to obtain anomaly scores for instances of source and target domain. As RVAE is trained with normal samples, the reconstruction errors of normal instances are expected to be lower than the reconstruction errors of anomalous instances. The more detailed discussion of Recurrent VAE can be referred in [8] and [24]. The details of the method to use RVAE as botnet detector are described in [16].

3 RELATED WORKS

Network IDS methods has been widely studied [1, 7, 13, 23, 25, 34]. In [1], the authors design a botnet decision engine which determines any divergence or statistical deviations, which are based on normal network behaviors, over network traffic data. However, the method cannot help failing to detect evolving botnets because it cannot detect new botnets. Above all, Zeek is one of the most popular Network Intrusion Detection System (IDS), which is a monitoring system for detecting network intruders in real-time by passively monitoring a network link over which the intruder's traffic transits [23]. Zeek analyzes `PCAP` file by utilizing `libpcap`, the packet-capture library. The system is divided into *event engine*, which reduces a stream of packets to a stream of higher-level network events, and *Policy Script Interpreter*, which logs real-time notifications and records data to disk. However, Zeek is not the IDS built for detecting botnet. Thus, the function of the software is not sufficient to be used as a botnet detection system.

The various ML methods aim to provide generalized botnet detection systems which are robust on previously unseen botnet [10, 19–21, 26, 28, 32]. Varialtional Autoencoder (VAE) and Autoencoder (AE) based methods have been proposed as semi-supervised learning technique for botnet detection [10, 16, 19, 20]. Recurrent Neural Network (RNN) based methods as supervised learning framework are utilized so as to consider periodicity of network traffic data [16, 21, 28, 31]. Moreover, other supervised learning anomaly detection methods such as Random Forest and Neural Network have been introduced in [11, 21, 32]. Although these methods provide impressive improvement on botnet detection, they usually require fully labeled dataset which is hard to obtain due to lack of labeled data on changing network traffic.

Some studies suggested making use of transfer learning on botnet detection [2, 6, 15, 17, 27, 30]. However, in [2], the method depends on naive techniques such as calculating similarity between each instance in the source and the target domain, which requires high computation cost. In [15], the authors use clustering technique and naive rule methods and only focus on the botnet using Command and Control channel. These methods are limited in that they cannot provide end-to-end learning manner while the proposed method can. Furthermore, contrary to transfer learning studies in other area which use Deep Neural Networks structure [4, 9, 14], the studies on botnet detection utilize relatively simple methods.

A few methods use Neural Networks in transfer learning [6, 27, 30]. The authors treat network traffic features as an image in [6, 30]. By bringing pre-trained Convolutional Neural Network (CNN) model which is suitable for image data, the authors do transfer learning to adapt network traffic data [6, 30]. Even though this approach is effective, the manners are quite different from the proposed method in that they use pre-trained parameters on image dataset. In [27], the authors propose transfer learning framework using Deep Neural Network(DNN), but this approach requires labeled dataset for both source and target domains contrary to the proposed method not requiring labeled dataset for a target domain. Moreover, the dataset that is used as source domains and target domains were generated on the same environment in [27].

4 PROPOSED MODEL

We propose a novel transfer botnet detection system which uses RVAE as an anomaly detector trained via transfer learning. From the anomaly detector, we can obtain anomaly scores given each instance which consists of reconstruction error from RVAE. As the anomaly detector is trained via transfer learning framework, each minibatch sample from a source domain and from a target domain is used for training, respectively and sequentially.

4.1 Anomaly Detection Method

The anomaly detector, RVAE, is given pre-processed flow-based features. These flow-based features are input in the order of time because the RNN model is sensitive to the order of the inputs. In the botnet detection system, the encoder is expected to be trained in a way of distilling the common characteristics within the sequential data into latent variable z. After that, the decoder reconstructs sequential inputs utilizing z. In the end, by comparing input with reconstructed input, we can obtain reconstruction errors and we use it as anomaly scores.

We train the model with only normal instances, and in evaluation phase, we calculate reconstruction errors using both normal and anomalous instances. As we only use normal samples for training, we expect that the reconstruction errors of anomalous samples are larger than that of the normal samples. For the anomaly detector, we collect each reconstruction loss in validation phase, then estimate distribution that represents collected reconstruction errors from normal and anomalous instances, respectively. In the testing phase, we get two likelihoods for each instance from normal and anomalous distributions. Ultimately, the network traffic flow data can be labeled by comparing the two values. The more detailed description of the botnet detection method can be referred in [16].

4.2 The Process of Transfer Learning

We follow the procedure of transfer anomaly detection method proposed in [17], but in order to adapt it to the characteristics of network traffic data, which are hard to obtain labeled data, we further develop the method which is able to be trained without label information on the target domain. Namely, we consider two cases of training data on botnet detection:labeled dataset on the target domain (*with_label*) and unlabeled dataset on the target domain (*without_label*).

In the both methods, normal and anomalous instances in a source domain are used for training RVAE at first. After calculating the gradients with a minibatch of samples of source domain and updating the parameters of the decoder and the encoder, we use a minibatch of samples of target domain for training. Then, we calculate and update the gradients in the same manner. The overall procedure of our proposed system is shown in Algorithm 1.

X_s^+ is a set of anomalous instances in a source domain ($x_s^+ \in X_s^+$). X_s^- is a set of normal instances in a source domain ($x_s^- \in X_s^-$). X_t^+ is a set of anomalous instances in a target domain ($x_t^+ \in X_t^+$). X_t^- is a set of normal instances in a target domain ($x_t^- \in X_t^-$). All elements belonged to those sets $x \in R^D$; and D is the number of features of network flow data. θ indicates parameters of the encoder of RVAE and F represents the encoder. Moreover, ϕ represents parameters of the decoder of RVAE and G represents the decoder. $\tilde{x}_n$ is nth reconstructed sequential input in the decoder of RVAE. z is the latent variable. N_s^+, N_s^- is the number of instances of anomalous and normal on the source domain.

We use the objective function of the source domain in [17]:

$$s_{\phi,\theta}(x|z) = \sum_{n=1}^{N} (1 - x_n) log(1 - \tilde{x}_n) + x_n log\tilde{x}_n \tag{1}$$

$$\begin{aligned} L_s(\theta, \phi|z) = & \frac{1}{N_s^-} \sum_{n=1}^{N_s^-} s_{\phi,\theta}(x_s^-|z) \\ & - \frac{\lambda}{N_s^- N_s^+} \sum_{n,m=1}^{N_s^-, N_s^+} f(s_{\phi,\theta}(x_s^+|z) - s_{\phi,\theta}(x_s^-|z)) \end{aligned} \tag{2}$$

$$\mathbb{L}_s(\phi, \theta) = \mathbb{E}_{q_\phi(z|x)}[L_s(\theta, \phi|z)] + \beta D_{KL}(q_\phi(z|X_s^-)|p(z)) \tag{3}$$

We did not use *latent domain vectors* suggested in [17] as there is only one domain for each source and target domain. Instead, we utilized VAE contrary to AE suggested in [17]. Thus, z represents latent variable in the proposed method.

The proposed method can be categorized into two depending on whether labeled dataset on the target domain is necessary or not. The overall process is identical, but transfer learning with unlabeled data set on the target domain is different from the method with the method using labeled data set on the target domain regarding that it uses entire instances in the target domain for training. While instances in the target domain are classified as normal and abnormal in *with_label* method, we do not distinguish between the normal instances and the anomalous instances in *without_label* method. Therefore, there is a slight difference in the objective function of the target domain of the two methods. On the contrary, the objective

functions for the source domain in both methods are identical as Equation 3.

4.2.1 using labeled data set in a target domain. In this case, we use only normal instances for training likewise other semi-supervised learning methods [10, 16, 20]. The difference from the semi-supervised learning is that we utilize transferred knowledge from the source domain to classify instances of the target domain. Therefore, we can expect better results than semi-supervised learning methods. The objective function for the target domain in *with_label* method is formulated as in [17]:

$$\mathbb{L}_t(\phi,\theta) = \mathbb{E}_{q_\phi(z|x)}\left[\frac{1}{N_t^-}\sum_{n=1}^{N_t^-} s_{\phi,\theta}(x_t^-|z)\right] + \beta D_{KL}(q_\phi(z|X_t^-)|p(z)) \tag{4}$$

The overall process of the method is shown in Algorithm 1. As you can see in Algorithm 1, we first utilize instances on the source domain. For training the model on the source domain, both abnormal and normal samples are required to get loss function in Equation 2. After updating the parameters of the model with the source domain dataset, we sample minibatch of the normal instances in the target domain, and then update the parameters. We iterate the process until all instances are utilized for training. However, the use of the method is limited in that the method requires only normal instances in the target domain. Namely, in order to obtain normal instances, we require the labeled data set for the target domain.

4.2.2 using unlabeled data set in a target domain. In this method, we assume the situation where there is no labeled data set on the target domain. That means we do not know what are normal instances and what are abnormal instances. To deal with such case, we use a entire instance of the dataset in the target domain during only for the first several epochs (E). From the very next sequence after E epochs, we collect instances which show lower reconstruction errors in each minibatch. We infer the instances with lower reconstruction errors have a higher probability to be normal because the number of normal samples is much higher than the number of anomaly. In order to give weight to the estimated normal instances, we use the instances more than once in the following minibatch training. In detail, we sort the instances by the size of reconstruction errors every minibatch. We then select instances of the bottom $r\%$ of reconstruction errors in minibatch as the estimated normal sample and add the bottom $r\%$ of instances to the following minibatch training samples. By utilizing the selecting samples method, we can train the anomaly detector effectively with unlabeled dataset on the target domain.

M_t is the number of the increased samples due to selection, and varies depending on the ratio (r). w_{x_t} is weights on each instance now that instances with lower reconstruction errors are used more than once. The objective function for target domain in *without_label* method is formulated:

$$\mathbb{L}_t(\phi,\theta) = \mathbb{E}_{q_\phi(z|x)}\left[\frac{w_{x_t}}{M_t}\sum_{n=1}^{M_t} s_{\phi,\theta}(x_t|z)\right] + \beta D_{KL}(q_\phi(z|X_t)|p(z)) \tag{5}$$

Algorithm 1: The Procedure of Training Transfer Anomaly Detection *with_label* Method

Input: instances of source domain $x_s^{-,+} \in X_s^{-,+}$ and instances of target domain $x_t^- \in X_t^-$
Output: G_θ, F_ϕ
Procedure
  for *the number of epochs* **do**
    Sample minibatches from X_s^+, X_s^- and X_t^- $(B_{x_s^-} \subset X_s^-, B_{x_s^+} \subset X_s^+, B_{x_t^-} \subset X_t^-)$
    for $B^a_{x_s^-}, B^c_{x_s^-}, B^e_{x_t^-}, (a = 1, ..., A, c = 1, ..., C, e = 1, ..., E)$ **do**
      forall $x_s^- \in B^a_{x_s^-}$ **do**
        $\tilde{x}_s^- = G_\theta(F_\phi(x_s^-))$
        forall $x_s^+ \in B^c_{x_s^-}$ **do**
          $\tilde{x}_s^+ = G_\theta(F_\phi(x_s^+))$
        end
        Update the Encoder and the Decoder by descending its stochastic gradient: $\nabla_{\theta,\phi}(L_s(\phi,\theta))$
      end
      forall $x_t^- \in B^e_{x_t^-}$ **do**
        $\tilde{x}_t^- = G_\theta(F_\phi(x_t^-))$
      end
      Update the Encoder and the Decoder by descending its stochastic gradient: $\nabla_{\theta,\phi}(L_t(\phi,\theta))$
    end
  end
end

5 EXPERIMENTS

We demonstrate the effectiveness of the proposed method using two network traffic datasets. The network log of two datasets are collected by Zeek which is a monitoring system for detecting network intruders in real-time [23].

We use the 5 different evaluation metrics to validate our performance; Area Under the Receiver Operating Characteristics (AUROC), True Positive rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR) and False Negative Rate (FNR). We save the model showing the best value of AUROC in validation set to avoid over-fitting. We use the mean of outputs each metric on the five identical experiments. The source code is written with the PyTorch[1] library.

5.1 Evaluation Datasets

The existing studies on transfer learning usually use the same dataset for target domains and a source domain [2, 6, 15, 17, 27, 30]. However, as our objective of the paper is detecting suspicious botnet connections on the new network monitoring dataset (target domain), we cannot help using target domain which is different from source domain since the data on the target domain cannot

[1] https://pytorch.org

be labeled. We use CTU-13 dataset as source domain which is widely used in the latest studies for botnet detection [3, 10–12, 19–21, 31]. CTU-13 dataset is labeled network traffic data. A botnet scenario is a particular infection of the virtual machines using a specific malware. Thirteen of these scenarios were created, and each of them was designed to be representative of some malware behavior [12]. In this paper, we only focus on botnet called Neris, which is used in scenario 1,2 and 9 in CTU-13 dataset, as in many existing studies [19, 21, 28]. Neris uses IRC protocols and sends SPAM. Also, it conducts port scanning and click fraud. We use whole data instances rather than separating *Normal* labels from *background* labels. The scenario 1,2 and 9 were collected for three days.

On the other hand, we use a network monitoring data set from a large research institute as a target domain dataset (called dataset K). The dataset is collected using a Zeek server connected at the network border. The Zeek server is installed all-in-one and used a default policy. We use the data collected for one day among seven days to balance the size of the target domain data with source domain data. The network monitoring data is not labeled. Even though every studies which use CTU-13 dataset utilize `Netflow` type of data which is provided in [12], we utilize data from Zeek software as dataset K is obtained from Zeek as well. In our experimental setting, the source domains and the target domain are different but related as we utilize two dataset generated on different environment. To sum up, we utilize CTU-13 Zeek dataset as a source domain, which labels are available, and use dataset K as a target domain, which labels are unavailable.

5.2 Labeling method

CTU-13 Zeek data has no label contrary to the original `Netflow` data. The dataset K has no labels neither. Therefore, new labeling method for the purpose of evaluating the model is necessary. We found labeling method that can provide the same strategy which can be applied to both CTU-13 and dataset K. Zeek's event engine record weird activity that can indicate malformed connections, malfunctioning or misconfigured hardware, or an attacker attempting to avoid/confuse a sensor. Also, Zeek provides specified type indication the reason why the connections provoke weird flags. Those suspicious connection are logged in `weird.log` in the system. However, we find that `weird.log` which is made by Zeek has no correlation with Botnet label in CTU-13 `Netflow` dataset. Many of the connections logged in `weird.log` might be made by misconfigured hardware or malformed connections, not made by Botnet. Moreover, most connections that are made by botnet in `Netflow` are not detected as "weird" activity in Zeek.

Nevertheless, we find that Neris accounts for 84% of connections with the indication of *irc_line_too_short* in `weird.log` among data from 13 scenarios. In addition, Neris accounts for 82% of connections with the indication of *irc_invalid_line* among data from 13 scenarios. We infer that most connections with the `weird.log` indication of *irc_line_too_short* and *irc_invalid_line* are given by Neris.

Therefore, we decide to use the indication information from `weird.log`, and label the host IP address with *irc_line_too_short* and *irc_invalid_line* as *malicious*. With the collected host IP addresses which are *malicious*, we can use network log features from `conn.log` composed of source and destination IP addresses and ports, time, protocol, duration, number of packets, number of bytes, state, and service. As both CTU-13 Zeek dataset and dataset K are made by Zeek, the labeling method can be applied commonly to two different datasets.

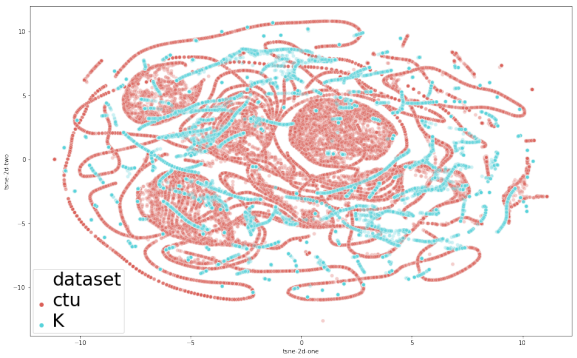


Figure 1: T-SNE plot over the source and the target domains

5.3 Data Preprocessing

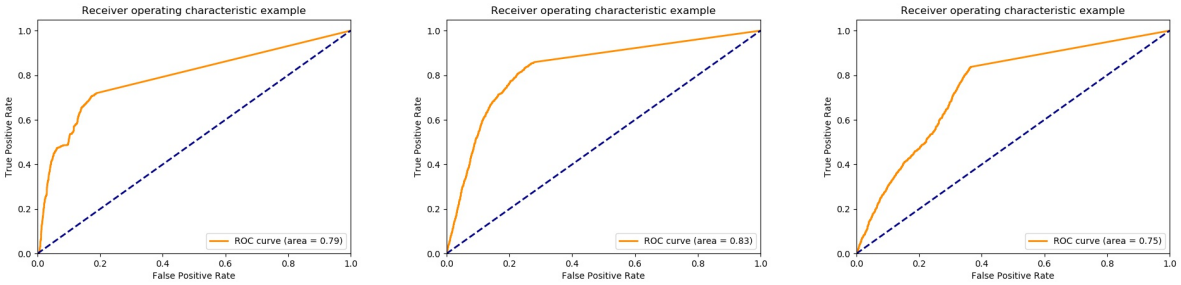
We follow the suggested data pre-processing method in [16]. We increase the number of features, as we add to use missed bytes and include more types of service such as *mysql*, *imap* and *ftp*. We also process the data to use the aggregated flows statistics as many existing studies [3, 10, 19–21, 31]. As a result, we obtain total 48 features that is larger than the number of features in [16].

5.4 Comparison Methods

We evaluated two variants of the proposed method: *with_label* and *without_label*. The more detailed description of the experimental setup of the proposed method such as hyperparameters can be referred in [16]. We compare those two suggested methods with RVAE which is a semi-supervised anomaly detection method in [16], which use only normal instances in the target domain for training.

6 RESULTS AND DISCUSSION

First, we plot t-SNE [18] of each instance of two datasets to justify to use transfer learning. To use transfer learning, two domains should be related and share common characteristics. We reduce dimension of features of data from 48 to 2 in order to visualize its distribution. The source domain dataset and the target domain dataset are not generated in the same environment. The source dataset, CTU-13 is made for the purpose of research for botnet detection in the environment where attacks of botnet are controlled. On the other hand, the target domain dataset K is network monitoring data which is collected using a Zeek server connected to the switch between the Internet and the local network. Therefore, the two distributions cannot be completely overlapping, as you can see in Fig. 1. However, because both data share common characteristics generated from Zeek, the two distributions are not completely separated. As a result, transfer learning, especially the application of *transductive transfer learning*, can show the improved performance over semi-supervised learning.



(a) RVAE AUROC **(b) *with_label* AUROC** **(c) *without_label* AUROC**

Figure 2: AUROC plot of one experiment over the target domain

Table 1: Average of each metric over the target domain (dataset K), r_s is 0.1 in the *without_label* method

Model	TNR	TPR	FNR	FPR	AUROC
RVAE	**0.685**	0.811	0.189	**0.309**	0.779
with_label	0.652	**0.915**	**0.084**	0.371	**0.810**
without_label	0.634	0.850	0.150	0.365	0.764

Second, we validate the proposed method by comparing with semi-supervised learning which uses RVAE method. We find that our proposed method *with_label* outperform RVAE, semi-supervised learning by large margins, as you can see in Table. 1 and Fig. 2. TPR, which is also called detection rate, of *with_label* method is 0.915 while TPR of RVAE method is 0.811 in Table. 1. Even we obtain higher detection rate with *without_label* method (0.850). Furthermore, the proposed method (*with_label*) show higher AUROC than the baseline as well. This output indicates that the effectiveness of using transfer learning as we use the same RVAE model and set the same hyper-parameters for training. Moreover, even *without_label* method which does not use label information on the target domain shows higher performance than RVAE on TPR and FNR metrics. Overall, these results demonstrate that the proposed method detects suspicious botnet better on the target domain as using transferred knowledge which is obtained on the related domain (source) can provide useful information for the target domain lack of training data.

7 CONCLUSION

We propose transfer learning framework as an effective botnet detection strategy. Transfer learning can learn on older data with labels and then apply the learning on new data records without label. This ability of working with unlabeled data is particularly useful for network security applications because security issues such as botnets continue to evolve. In our tests, we train neural network on labeled data from CTU-13 and apply the network for anomaly detection on a fresh set of network monitoring data. Tests show that transfer learning could reliably identify anomalies. The accuracy as measured by area under the curve for transfer learning is higher than a sophisticated semi-supervised learning based method trained on the target data set (0.810 vs 0.779).

For future studies, we plan to study some improvements in the proposed method. First, we propose an empirical manner of using transfer anomaly detection method without labels on a target domain. Future research will be required to propose more systematic method beyond empirical ways to improve *without_label* method. In addition, it is potential to improve performance of the anomaly detector in FPR measure as it show weak performance relatively. Moreover, the RVAE architecture can be replaced with other ML methods to improve its anomaly detection performance.

ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and also used resources of the National Energy Research Scientific Computing Center (NERSC). The authors also gratefully acknowledge Kimoon Jeong of National Supercomputing & Networking, Korea Institute of Science and Technology Information (KISTI).

REFERENCES

[1] Abbas Abou Daya, Mohammad A Salahuddin, Noura Limam, and Raouf Boutaba. 2020. BotChase: Graph-Based Bot Detection Using Machine Learning. *IEEE Transactions on Network and Service Management* (2020).
[2] Basil Alothman. 2018. Similarity-Based Instance Transfer Learning for Botnet Detection.
[3] Jinwon An and Sungzoon Cho. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE* 2, 1 (2015).
[4] J Andrews, Thomas Tanay, Edward J Morton, and Lewis D Griffin. 2016. Transfer representation-learning for anomaly detection. JMLR.
[5] Elaheh Biglar Beigi, Hossein Hadian Jazi, Natalia Stakhanova, and Ali A Ghorbani. 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In *2014 IEEE Conference on Communications and Network Security*. IEEE, 247–255.
[6] Niket Bhodia, Pratikkumar Prajapati, Fabio Di Troia, and Mark Stamp. 2019. Transfer learning for image-based malware classification. *arXiv preprint arXiv:1903.11551* (2019).
[7] James R Binkley and Suresh Singh. 2006. An algorithm for anomaly-based botnet detection. *SRUTI* 6 (2006), 7–7.
[8] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).
[9] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. 2018. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360* (2018).
[10] Ruggiero Dargenio, Shashank Srikant, Erik Hemberg, and Una-May O'Reilly. 2018. Exploring the Use of Autoencoders for Botnets Traffic Representation. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 57–62.

[11] Fei Du, Yongzheng Zhang, Xiuguo Bao, and Boyuan Liu. 2019. FENet: Roles Classification of IP Addresses Using Connection Patterns. In *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 158–164.

[12] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. 2014. An empirical comparison of botnet detection methods. *computers & security* 45 (2014), 100–123.

[13] Guofei Gu, Junjie Zhang, and Wenke Lee. 2008. BotSniffer: Detecting botnet command and control channels in network traffic. (2008).

[14] Tsuyoshi Idé, Dzung T Phan, and Jayant Kalagnanam. 2017. Multi-task multi-modal models for collective anomaly detection. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 177–186.

[15] Jianguo Jiang, Qilei Yin, Zhixin Shi, Meimei Li, and Bin Lv. 2019. A New C&C Channel Detection Framework Using Heuristic Rule and Transfer Learning. In *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–9.

[16] Jeeyung Kim, Alex Sim, Jinoh Kim, and Kesheng Wu. 2020. Botnet Detection Using Recurrent Variational Autoencoder. arXiv:2004.00234

[17] Atsutoshi Kumagai, Tomoharu Iwata, and Yasuhiro Fujiwara. 2019. Transfer Anomaly Detection by Inferring Latent Domain Representations. In *Advances in Neural Information Processing Systems*. 2467–2477.

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[19] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. 2019. GEE: A gradient-based explainable variational autoencoder for network anomaly detection. In *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 91–99.

[20] Miguel Nicolau, James McDermott, et al. 2018. Learning neural representations for network anomaly detection. *IEEE transactions on cybernetics* 49, 8 (2018), 3074–3087.

[21] Talha Ongun, Timothy Sakharaov, Simona Boboila, Alina Oprea, and Tina Eliassi-Rad. 2019. On Designing Machine Learning Models for Malicious Network Traffic Classification. *arXiv preprint arXiv:1907.04846* (2019).

[22] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[23] Vern Paxson. 1999. Bro: a system for detecting network intruders in real-time. *Computer networks* 31, 23-24 (1999), 2435–2463.

[24] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428* (2018).

[25] Martin Roesch et al. 1999. Snort: Lightweight intrusion detection for networks.. In *Lisa*, Vol. 99. 229–238.

[26] Kamaldeep Singh, Sharath Chandra Guntuku, Abhishek Thakur, and Chittaranjan Hota. 2014. Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences* 278 (2014), 488–497.

[27] Ankush Singla, Elisa Bertino, and Dinesh Verma. 2019. Overcoming the Lack of Labeled Data: Training Intrusion Detection Models Using Transfer Learning. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 69–74.

[28] Kapil Sinha, Arun Viswanathan, and Julian Bunn. 2019. Tracking Temporal Evolution of Network Activity for Botnet Detection. *arXiv preprint arXiv:1908.03443* (2019).

[29] Matija Stevanovic and Jens Myrup Pedersen. 2014. An efficient flow-based botnet detection using supervised machine learning. In *2014 international conference on computing, networking and communications (ICNC)*. IEEE, 797–801.

[30] Shayan Taheri, Milad Salem, and Jiann-Shiun Yuan. 2018. Leveraging Image Representation of Network Traffic Data and Transfer Learning in Botnet Detection. *Big Data and Cognitive Computing* 2, 4 (2018), 37.

[31] Pablo Torres, Carlos Catania, Sebastian Garcia, and Carlos Garcia Garino. 2016. An analysis of recurrent neural networks for botnet detection behavior. In *2016 IEEE biennial congress of Argentina (ARGENCON)*. IEEE, 1–6.

[32] G Kirubavathi Venkatesh and R Anitha Nadarajan. 2012. HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network. In *IFIP International Workshop on Information Security Theory and Practice*. Springer, 38–48.

[33] Yanshan Xiao, Bo Liu, S Yu Philip, and Zhifeng Hao. 2015. A robust one-class transfer learning method with uncertain data. *Knowledge and Information Systems* 44, 2 (2015), 407–438.

[34] Hossein Rouhani Zeidanloo, Mohammad Jorjor Zadeh Shooshtari, Payam Vahdani Amoli, M Safari, and Mazdak Zamani. 2010. A taxonomy of botnet detection techniques. In *2010 3rd International Conference on Computer Science and Information Technology*, Vol. 2. IEEE, 158–162.

[35] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, and Dan Garant. 2013. Botnet detection based on traffic behavior analysis and flow intervals. *Computers & Security* 39 (2013), 2–16.