A Novel Transfer Learning Approach for Detecting Unseen Anomalies

Khan Mohammad Al Farabi
University of Georgia
kma27987@uga.edu

Gagan Agrawal
University of Georgia
gagrawal@uga.edu

***Abstract*—The identification of anomalies and outliers through machine-learning models constitutes a critical and extensively studied domain. However, lack of training data constitutes a significant concern in many application domains. Considering the realm of network intrusion detection – detecting previously unseen attacks, and/or attacks on devices for which training data has not been collected – is essential for implementing effective security measures. To address this issue, we propose an innovative transfer-learning approach. Our method generates new labels for source datasets by leveraging insights from the target dataset, thereby facilitating the accurate identification of normal data within the source datasets. Subsequently, we train an unsupervised model using these normal data to infer unseen anomalies. Furthermore, we employed state-of-the-art explainable artificial intelligence techniques to extract significant features from source datasets, thereby enhancing our model's performance. Our comprehensive evaluations clearly demonstrate the efficacy of our method across both generic (non-IoT) and IoT datasets, consistently surpassing existing transfer learning approaches. For example, our method achieved a remarkable accuracy of 0.99 in detecting anomalous traffic from UDP flooding attacks, whereas the well-known transfer learning model, SSkNNO, achieved an accuracy of only 0.83.**

***Index Terms*—Transfer Learning, Anomaly Detection, IoT Attacks.**

I. Introduction

Anomaly detection, also known as outlier detection, is a fundamental task in data analysis [1]. Data that deviate from the norm are identified as anomalous and detecting such anomalous data in real-world datasets presents significant challenges. One particular challenge that has received relatively less attention is that in many application domains, sufficient training data is not available. Accurately labeling them necessitates human intervention, which is both costly and time consuming.

To further motivate this, we consider one of the popular application area for anomaly detection, network intrusion detection [2]. IoT devices are threatened by various attacks that compromise their security. The key observation that has allowed the detection of such attacks is that the behavior of the network traffic changes due to these attacks. For example, HTTP flood attacks [3] send a large number of HTTP requests to the server, causing overload on the network and an inaccessible website. SYN DDoS attacks [4] send a large number of SYN messages to ports (destination or source ports) making the connection of IoT devices unreliable.

To identify anomalous network traffic, traditional supervised algorithms require labeled data for training. However, finding training data for all cases is not feasible. Well-known unsupervised anomaly detection models such as one-class SVM [5] do not require labeled data to detect anomalous traffic. The one-class SVM model depends on the nearest-neighbor-based technique and considers the traffic data as anomalies that are not near its neighbors. However, these state-of-the-art models have not been shown to exhibit good performance in detecting unseen attacks without appropriate normal data information and training data.

The requirement for data labeling can be alleviated by employing transfer learning approaches. Transfer learning [6] has recently become a mechanism for acquiring knowledge from one domain and applying it to related domains. In our specific case, the advantage of transfer learning approaches is that they can efficiently identify unseen anomalous data without requiring the specific labeled data of an attack or network protocol. The most advanced transfer learning methods are mostly *semi-supervised* [7] – that is, a small amount of ground truth data is needed to efficiently apply these semi-supervised models. However, current studies do not address the cases where attacks are completely unseen and the data related to a new attack are completely unlabeled. In developing a completely unsupervised method, the challenge is to extract the transfer knowledge and utilize it to detect unseen anomalous network traffic data of the affected IoT devices.

This paper develops a method where target dataset labels are not required. The crux of our approach is in relabeling source dataset records using two new approaches. The first approach is that a one-class SVM model is trained on the *normal* records from the target dataset, where records have been classified as normal or abnormal based on an anomaly detection method. The second approach classifies the data points in source dataset as abnormal based on average distance from the anomalous points in the target dataset. Voting using three labels, i.e., the original labels and the two approaches above, gives the new labels to source dataset. These new labels, together with the selection of the most important features using an Explainable AI (XAI) [8] tool, provide the normal data with prominent features of the source dataset to develop our final inference model.

Our experimental results demonstrate that our approach outperforms unsupervised approaches: one-class SVM (without

979-8-3315-6689-0/25/$31.00 ©2025 IEEE
DOI 10.1109/ICKG66886.2025.00008

transfer learning) and other state-of-the-art approaches (with transfer learning). For example, our approach achieves an average accuracy of 0.99 in detecting unseen HTTP attacks. In contrast, the average accuracy values for the baseline approaches: One-Class SVM (OCS) [9], Local Outlier Factors (LOF) [10], Isolation Forest (ISOF) [11], One-Class SVM Stochastic Gradient Descent (OCS-SGD) [12], isolation nearest neighbor ensembles (iNNE) [13], k-nearest neighbor outlier detection (knno) [14], and Robust Covariance (RC) [15] are 0.79, 0.69, 0.14, 0.81, 0.84, 0.81, and 0.46, respectively. Similarly, for the generic dataset (non-IoT data), the F-1 score for our approach is 0.98, whereas the F-1 scores for knno and iNNE are 0.74 and 0.95 respectively. Our method significantly surpasses these state-of-the-art techniques.

II. Proposed Approach

We propose an approach that efficiently infers the unseen anomalous items without the requirement of the labeled data. As in any transfer learning scenario, we have *source* and *target* domains. The source data set is labeled, while the target data set is not. The source domain contains both normal and abnormal traffic data, where abnormal traffic data points are affected by *known attacks*. The target domain contains a combination of unlabeled normal and abnormal traffic data points where the abnormal data are affected by unknown attacks. For example, we demonstrate the real-time traffic data from the source and target domains in Figure 2. The traffic data for the source domain with the original labels are shown in Figure 2 (a). Figure 2 (b) shows the unlabeled traffic data of the target domain. In general, data points from the target domain can involve a feature set that differs from that of the source domain.

Successful anomaly depends on determining the accurate characteristics of normal behavior. Like the current transfer learning anomaly detection methods [16], we did not transfer the labeled data from *source* to *target* domain to develop inference models. The key idea in our method is to use information from target dataset to provide additional labels for source dataset, and then use voting to decide on new labels for source data. Then, we leverage these newly labeled source data to enable the XAI model to extract the important features of the source data. This is followed by developing a new model on source data with prominent features and then applying it to the target dataset to infer the unseen attacks. We present the pipeline of our approach in Figure 1.

A. Detailed Methodology

In this part, we illustrate the steps involved in implementing our method using an example.

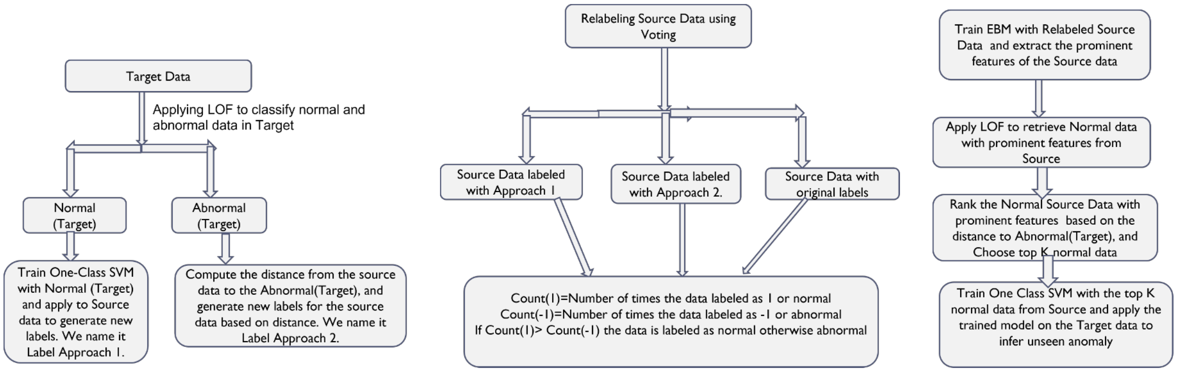
New Labeling Approach 1: We leverage unsupervised anomaly detection algorithms: Local Outlier Factor (LOF) and one-class SVM model to leverage unlabeled traffic data in the target domain. We train LOF with the unlabeled traffic data in target domain, and infer the normal traffic from the target domain by using the trained LOF. We leverage LOF because it efficiently detects outlier data based on density. LOF successfully filters out the normal data points from the target domain. Then we train the one-class SVM model with these normal (non-anomalous) traffic data. Next, we apply the trained one-class SVM model to infer the first set of new labels for source data, i.e. whether the traffic data point in the source domain is a normal data point, independent of the existing labels. Our approach provides the link between the source domain and target domain – in other words, this is the transfer learning phase where the learnt information from target domain is utilized in source domain to extract normal traffic data.

New Labeling Approach 2: In the second phase, we determine the cosine distance between data points in the source and the anomalous data points in the target domain (LOF is used to detect anomalous data points in the target domain as before). We have applied Principal Component Analysis (PCA) to obtain the same feature space for network traffic data in the source domain and the target domain. PCA helps to have the same dimensions for the network traffic data in the source domain and the target domain. We require the same dimension of the data to compute the cosine distance. Next, for each data point in the source domain, the cosine distances to all the anomalous data points in the target domain are calculated. The cosine distance is within the range of 0–1. We consider a data point of the source domain to be closer to the anomalous traffic data points of the target domain when the average distance of that data point to the set of anomalous points is less than 0.20. Such a data point is labeled as abnormal or anomalous.

As shown in Figure 2, [43, 36, 66] is the outlier data of the target domain. The cosine distances for the source domain's traffic data to the target domain's outlier data are computed, and in Figure 3 we have shown how the data points of the source domain are labeled based on the distance. Records [87, 78, 17] and [74, 24, 16] data in source domain are labeled as normal because the distances to the target domain's outlier ([43, 36, 66]) are 0.82, and 0.88 respectively. However, the last data point has a distance of 0.16 and is labeled as abnormal.

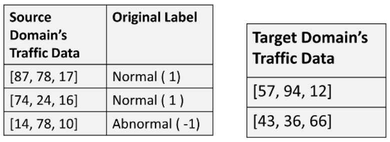
Voting: To use both the sets of additional labels computed above, as well as the original labels in source domain, a voting method is used. A data point is considered normal point in the source domain if at least two of the three labels are normal, otherwise, it is considered abnormal. The rest of the steps in our algorithms use the new labels assigned in the source domain. In Figure 4, we have shown our voting process where the records [87, 78, 17], and [74, 24, 16] in the source domain are labeled as normal because the normal label has the highest vote. The record [14, 78, 10] data is labeled as abnormal as the abnormal label has the highest vote.

Source Domain's Feature Selection: As a background, Explainable Artificial Intelligence (XAI) [8] methods detect important features of data to explain the inference outcomes. The prominent XAI tool, Explainable Boosting Machine (EBM) [17] can be used to extract the prominent features, and we leveraged the EBM model to extract features of the benchmark data. EBM model is inherently interpretable, and transparent in important features extraction. EBM is also an *glassbox* approach that provides both *global* and *local*


Target Data
Applying LOF to classify normal and abnormal data in Target
Normal (Target)
Abnormal (Target)
Train One-Class SVM with Normal (Target) and apply to Source data to generate new labels. We name it Label Approach 1.
Compute the distance from the source data to the Abnormal(Target), and generate new labels for the source data based on distance. We name it Label Approach 2.
Relabeling Source Data using Voting
Source Data labeled with Approach 1
Source Data labeled with Approach 2.
Source Data with original labels
Count(1)=Number of times the data labeled as 1 or normal
Count(-1)=Number of times the data labeled as -1 or abnormal
If Count(1)> Count(-1) the data is labeled as normal otherwise abnormal
Train EBM with Relabeled Source Data and extract the prominent features of the Source data
Apply LOF to retrieve Normal data with prominent features from Source
Rank the Normal Source Data with prominent features based on the distance to Abnormal(Target), and Choose top K normal data
Train One Class SVM with the top K normal data from Source and apply the trained model on the Target data to infer unseen anomaly

(a) (b) (c)

Fig. 1. Flow-chart Showing Our Approach



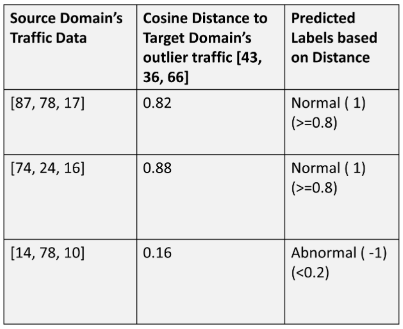
Source Domain's Traffic Data	Original Label
[87, 78, 17]	Normal (1)
[74, 24, 16]	Normal (1)
[14, 78, 10]	Abnormal (-1)

(a) Source Domain

Target Domain's Traffic Data
[57, 94, 12]
[43, 36, 66]

(b) Target Domain

Fig. 2. Source Domain network traffic in (a), and Target Domain network traffic data in (b).

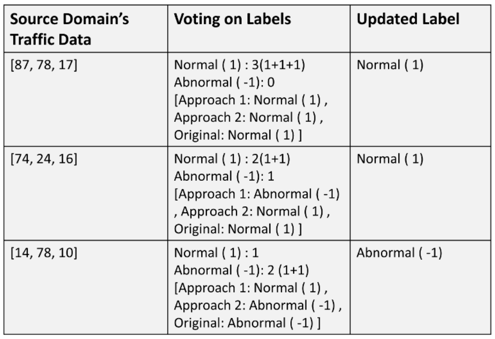


Source Domain's Traffic Data	Cosine Distance to Target Domain's outlier traffic [43, 36, 66]	Predicted Labels based on Distance
[87, 78, 17]	0.82	Normal (1) (>=0.8)
[74, 24, 16]	0.88	Normal (1) (>=0.8)
[14, 78, 10]	0.16	Abnormal (-1) (<0.2)

Fig. 3. New Source Dataset Labeling Based on Distance from Anomalous Points in Target Dataset (Approach 2).

explanations to justify the inference outcomes. Here, the global explanations define the contributions of the features in making the model's decision, and represent the prominent features of the data. This includes the scores of the features computed by the cumulative summation of the contribution weights of the feature in the inference for the entire dataset. Local explanations, in comparison, represent the features that have the most significant influence on predicting an individual instance.

With new labels on the source data, we train the EBM model – specifically, we leverage *explain global* function to retrieve the global explanations of EBM model. The global explanations provides the features with influential weights [17]. We rank the features based on influential weights computed using



Source Domain's Traffic Data	Voting on Labels	Updated Label
[87, 78, 17]	Normal (1) : 3(1+1+1) Abnormal (-1): 0 [Approach 1: Normal (1) , Approach 2: Normal (1) , Original: Normal (1)]	Normal (1)
[74, 24, 16]	Normal (1) : 2(1+1) Abnormal (-1): 1 [Approach 1: Abnormal (-1) , Approach 2: Normal (1) , Original: Normal (1)]	Normal (1)
[14, 78, 10]	Normal (1) : 1 Abnormal (-1): 2 (1+1) [Approach 1: Normal (1) , Approach 2: Abnormal (-1) , Original: Abnormal (-1)]	Abnormal (-1)

Fig. 4. Updated labels of the Source Domain's Data based on Voting.

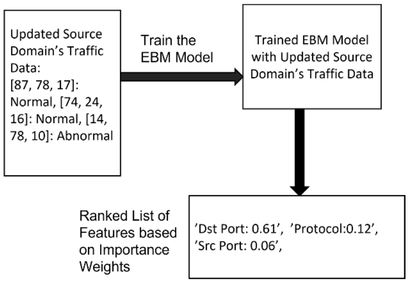

Updated Source Domain's Traffic Data: [87, 78, 17]: Normal, [74, 24, 16]: Normal, [14, 78, 10]: Abnormal
Train the EBM Model
Trained EBM Model with Updated Source Domain's Traffic Data
Ranked List of Features based on Importance Weights
'Dst Port: 0.61', 'Protocol:0.12', 'Src Port: 0.06',

Fig. 5. Ranked feature extraction from Source Domain using EBM model.

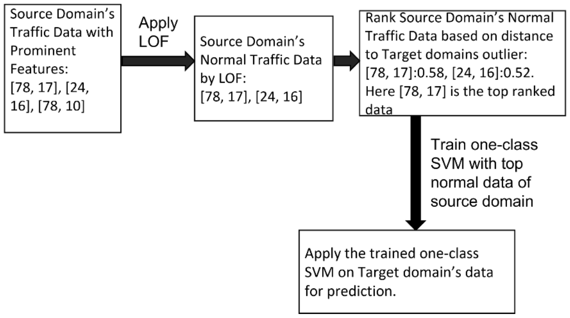

Source Domain's Traffic Data with Prominent Features: [78, 17], [24, 16], [78, 10]
Apply LOF
Source Domain's Normal Traffic Data by LOF: [78, 17], [24, 16]
Rank Source Domain's Normal Traffic Data based on distance to Target domains outlier: [78, 17]:0.58, [24, 16]:0.52. Here [78, 17] is the top ranked data
Train one-class SVM with top normal data of source domain
Apply the trained one-class SVM on Target domain's data for prediction.

Fig. 6. Inference of Anomalous Traffic.

the EBM model. We select the top k features. We leverage only k features, where k is a preset parameters, are used for subsequent analysis. Figure 5 shows the steps involved in the prominent feature extraction process for network traffic data from the source domain. Next, we choose top two prominent features. For further analysis, only these features are considered in this example. From Figure 5 we see 'Dst Port', and 'Protocol' are the important features of the traffic data from the source domain.

Inferring Anomalous Records: We utilize not just the prominent features but also the *most-distinct* normal traffic data from the source domain to train an unsupervised one-class SVM model. We first train LOF with the data from source domain, and next we apply this trained LOF on the source data to filter out the normal data. We extract the normal data from source domain that are labeled as normal or 1 by LOF. To find the most distinct one, the ranking of normal traffic data is done as follows. The cosine distance is calculated between the data points of the target domain and the normal traffic data of the source domain. Next, we find the average distance for each normal traffic data point in the source domain to the data points of the target domain. We sort the normal traffic data of the source domain by average distance in descending order and choose the highest-ranked data that differ from the target domain's traffic data. Next, we train the unsupervised one-class SVM model with these highest-ranked normal data from the source domain. This trained model is then used to detect anomalous traffic in the target domain. Figure 6 demonstrates the inference of the anomalous traffic in the target domain. We did not use the traffic data from the target domain in this training phase to avoid bias and data leakage. Our approach involves the prominent feature selection process for the source domain only which is influenced by the learnt information from the target domain. Our approach can efficiently infer the anomalous traffic of the target domain that have completely different features than that of the source domain. We have shown the efficacy of our approach through empirical analysis.

III. Experimental Results

We conducted a series of experiments with the primary goal of comparing our approach with the prominent state-of-the-art anomaly or outlier detection models, with respect to accuracy and the F-1 score. This evaluation uses both datasets from the IoT domain, and other more general datasets previously used in research on transfer learning.

A. Baseline Approaches

We empirically compared our approach with the following approaches, including a set of transfer learning approaches and other unsupervised models for anamoly detection.

Unsupervised Models: We used unsupervised algorithms that do not leverage transfer learning: One-Class SVM (OCS) [9], Local Outlier Factors (LOF) [10], Isolation Forest (ISOF) [11], One-Class SVM Stochastic Gradient Descent (OCS-SGD) [12], and Robust Covariance (RC) [15]. OCS, RC, and OCS-SGD are distance-based unsupervised approaches to detect the anomaly traffic, whereas LOF, is a density based local outlier detector model. ISOF is based on decision tree [18]. We also compared with kNNO (k-nearest neighbor outlier detection), which is is an unsupervised distance-based outlier detection method that leveraged KNN classifier [19]. The last unsupervised method we compare against is iNNE [13] (isolation nearest neighbor ensembles), which is nearest-neighbor ensemble based unsupervised method.

Semi-supervised Methods: We used SSDO [20] (semi-supervised detection of outliers), which computes the anomaly score of an unknown instance by utilizing both transferred labeled data and unlabeled data. We also used SSkNNO [16], which stands for semi-supervised k-nearest neighbor anomaly detection. We also compared our approach with Semi-Supervised Isolation Forest method [21].

B. Datasets

As stated earlier, two sets of datasets were used in our work.

Generic Datasets: We have obtained seven non-IoT data sets from the extensive research conducted by Vincent et al. [16]. We refer to these datasets as *generic* datasets. These datasets encompass: Gas Sensor, Gesture Segmentation, Handwritten Digits, Landsat Satellite, Letter Recognition, Shuttle, and Waveform. Each dataset includes both source and target data. The number of records in the source datasets are: 5,159 for Gas Sensor, 4,685 for Gesture Segmentation, 1,451 for Handwritten Digits, 2,971 for Landsat Satellite, 1,818 for Letter Recognition, 9,197 for Shuttle, and 2,800 for Waveform. The number of records in the target datasets are as follows: 1,800 for Gas Sensor, 1,600 for Gesture Segmentation, 460 for Handwritten Digits, 880 for Landsat Satellite, 600 for Letter Recognition, 4,800 for Shuttle and 1,100 for Waveform.

IoT Datasets: We evaluated the proposed approach using three IoT datasets. The first dataset was collected internally by our collaborators. The data collection leveraged the well-known packet capture tool, Wireshark [22] to capture network traffic data from seven IoT devices: Echo Dot, Google Home CAM, Google Nest Mini, Kasa, LongPlus Smart PTZ, Nite Bird Smart LED, and Ring Doorbell. Flood attacks such as TCP [4], UDP [23], XMAS [24], and HTTP [3] were applied to these IoT devices. Subsequently, both normal and affected (abnormal) network traffic data from these devices for each flood attack was obtained. We leveraged 79 numerical features and 2000 network traffic packets per device for an attack in our experiment. The label information was used only as ground truth and not for any training. We have shared this dataset with a GitHub link. The second dataset is Kitsune [25], which contains network traffic data affected by nine different attacks: ARP MitM, SSDP Flood, OS Scan, Active Wiretap, SYN DoS, Fuzzing, Video Injection, SSL Renegotiation, and Mirai malware. We used all 115 features and nearly a million packets for each attack in our experiment. The third dataset is the IoT-23 dataset [26]. The abnormal traffic data here is labeled with twelve different attacks. We used three million packets along with nine numerical features. We refer to our first dataset as "OD", the second dataset as "SD", and the third

TABLE I
COMPARISON BETWEEN THE PROPOSED APPROACH AND SEMI-SUPERVISED MODELS WITH RESPECT TO F-1 SCORE FOR SEVEN NON-IOT DATASETS.

Datasets	Proposed Approach	SSIF	iNNE	kNNO	SSkNNo	SSDO
Gas Sensor	**0.99**	0.93	0.89	0.82	0.83	0.49
Gesture Segmentation	**0.98**	0.96	0.87	0.90	0.91	0.92
Handwritten Digits	**0.97**	0.94	0.92	0.75	0.64	0.88
Landsat satellite	**0.99**	0.92	0.93	0.92	0.77	0.62
Letter recognition	**0.98**	0.94	0.96	0.94	0.79	0.94
Shuttle	**0.99**	0.95	0.97	0.98	0.63	0.95
Waveform	**0.98**	0.88	0.95	0.74	0.97	0.80

dataset as "IoT-23" in this section. In our implementations, we leveraged the top ten important features (extracted by EBM in our method) for the datasets "OD", and "SD". For the third dataset "IoT-23" we also utilize the EBM model to identify the five most significant features for inclusion in our empirical analysis.

C. Evaluation Results for Generic Datasets

In this section, we have compared our approach with other state-of-the-art transfer learning approaches for generic datasets. We have shown that our approach is applicable in general dataset with high performance to detect the unseen anomalies, and hence proves the generalization of our approach efficiently.

As previously stated, we successfully collected seven datasets from the comprehensive study conducted by Vincent et al. [16]. To eliminate any potential bias, we ensured that the training and test data sizes were consistent across all methods. Our performance evaluation was grounded in the F-1 score for this experiment, and we selected the top 100 high-ranked data where the data are ranked based on the distance to the abnormal data in *target* domain (as we described in previous section) for training from each of the seven datasets for our approach. The results in Table I clearly demonstrate the strength of our approach, achieving impressive F-1 scores: 0.99 for Gas Sensor, 0.98 for Gesture Segmentation, 0.97 for Handwritten Digits, 0.99 for Landsat Satellite, 0.98 for Letter Recognition, 0.99 for Shuttle, and 0.98 for Waveform.

For the SSIF (Semi-Supervised Isolation Forest) method [21], we randomly selected 100 data points for training and computed the F-1 scores for anomaly predictions. This experiment was carried out 500 times to obtain robust average F-1 scores for each dataset. For each case, we make sure training data are different from the test data to avoid bias. The resulting scores are 0.93, 0.96, 0.94, 0.92, 0.94, 0.95, and 0.88 for the Gas Sensor, Gesture Segmentation, Handwritten Digits, Landsat Satellite, Letter Recognition, Shuttle, and Waveform datasets respectively. Similarly, for the SSkNNo method [16], the average F-1 scores over 500 iterations were calculated, yielding results of 0.83, 0.91, 0.64, 0.77, 0.79, 0.63, and 0.97 for the Gas Sensor, Gesture Segmentation, Handwritten Digits, Landsat Satellite, Letter Recognition, Shuttle, and Waveform datasets respectively.

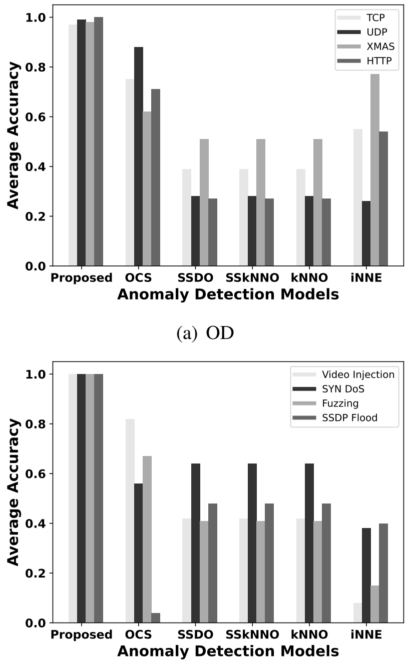
Analyzing Table I, it is evident that our approach (0.98) outperforms the SSkNNO method (0.97) in the Waveform dataset. Additionally, in the Gesture Segmentation dataset, the SSIF method (0.96) came strikingly close to our approach (0.98). Subsequently, we have shown that our approach beats the kNNO, iNNE, and SSDO methods across all the generic datasets concerning F-1 scores. Table I consolidates the results, clearly indicating that our method consistently surpasses the SSIF, SSkNNO, kNNO and iNNE methods regarding F-1 scores in detecting unseen anomalies.

D. Evaluation Results for IoT Datasets

In this section, we have shown the efficacy of our approach in detecting the unseen anomalies in the network traffic data and also outperforms the current transfer learning state-of-the-art methods.

Transfer Learning Across Attacks: In this experiment, we evaluated the effectiveness of transfer learning across attacks. For our first dataset, we had four flood attacks: TCP, UDP, XMAS, and HTTP. We performed a four-fold cross-validation of these four attacks, i.e., we generate training data by using traffic data affected by three attacks and then test it using the fourth attack. For each of the four such cases, we randomly took 250 network traffic packets for training and 500 network traffic packets for testing for the first dataset "OD". We repeated the experiment ten times, and computed the average accuracy. Figure 7(a) shows the results for the dataset "OD" . Although OCS shows better performance in detecting anomalous traffic affected by UDP attack, our approach outperforms OCS in terms of average accuracy. Similarly, our approach significantly outperforms other state-of-the-art approaches in detecting anomalous traffic. For the second dataset "SD" we choose four different attacks, i.e., Video Injection [25], SYN DoS [25], Fuzzing [27], and SSDP Flood [25] and perform the four-fold cross validation experiment as the above. The comparison of average accuracy values between our approach and other methods is shown in Figure 7(b). We used 500 random traffic packets as training data, and leveraged 1000 traffic packets for testing. We repeated the experiment for ten times and computed average accuracy. In detecting the anomalous traffic affected by Video Injection attack, OCS shows better performance compared to SSDO [20], SSkNNO [16], kNNO, and iNNE method. Nevertheless, our approach beats OCS in terms average accuracy in detecting traffic affected by Video Injection attack. Similarly, we observe our approach outperforms the baseline methods significantly in detecting other attacks.

Transfer Learning Across IoT Devices' Traffic: In this experiment, we generated training and test data by varying the IoT traffic devices. The first dataset was used in this experiment. Training data was generated by varying the number of IoT devices used for TCP, UDP, and XMAS flood attacks. We randomly selected five out of seven devices and generated training data – the specific choice of these was varied and averages were taken. Next, we utilized the traffic data of the remaining two devices as the test data. We used 100 random



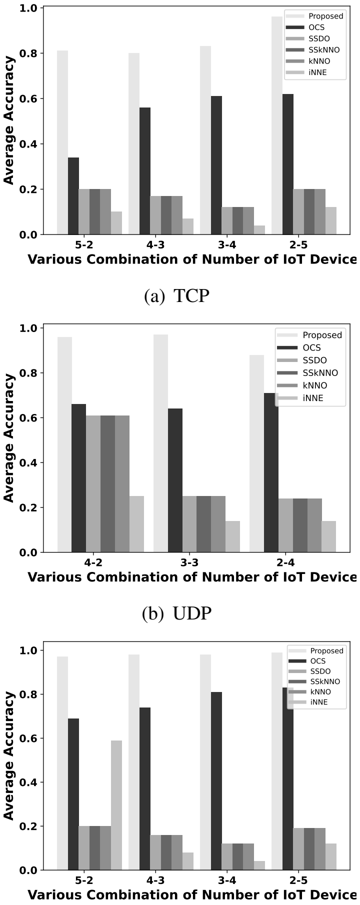
(a) OD

(b) SD

Fig. 7. Comparison (average accuracy) between the proposed approach and others approaches: transfer learning across attacks: dataset OD (a) and SD (b).

data points for training and 500 traffic packets for testing. This experiment was repeated five times and the average accuracy was calculated. Figure 8 (a), (b), and (c) show comparison between our approach and other state-of-the-art baseline methods for TCP, UDP, and XMAS flood attacks, respectively. We observe that, unsupervised approach, OCS again beats SSDO, SSkNNO, kNNO, and iNNE methods in detecting anomalous traffic affected by TCP, UDP, and XMAS flood attacks. The results demonstrate that our approach outperforms OCS, and other baseline state-of-the-art methods significantly with respect to accuracy. Particularly, the prominent feature selection and accurate normal traffic packet detection make our approach more robust in determining anomalous traffic.

Transfer Learning Across Datasets: The purpose of this experiment was to verify whether the model can detect completely unseen abnormal network traffic. Here, the unseen traffic means that the network traffic has different features than the network traffic data used in the training. Furthermore, these unseen network traffic data are affected by unknown attacks that differ from those in the training data. In this experiment, we used “OD” for generating training data and used “SD” for the test data. We extracted traffic packets for Video Injection [25], SYN DoS [25], Fuzzing [25], and SSDP Flood [25] attacks in “SD”. Next, we integrated the traffic data affected by flood attacks (TCP, UDP, XMAS, and HTTP) from “OD”, and randomly selected 100 traffic data points for the training. Then for each of the fours attacks in “SD”, we collected 1000 traffic packets for testing. We repeated the experiment for ten runs and computed the average accuracy.



(a) TCP

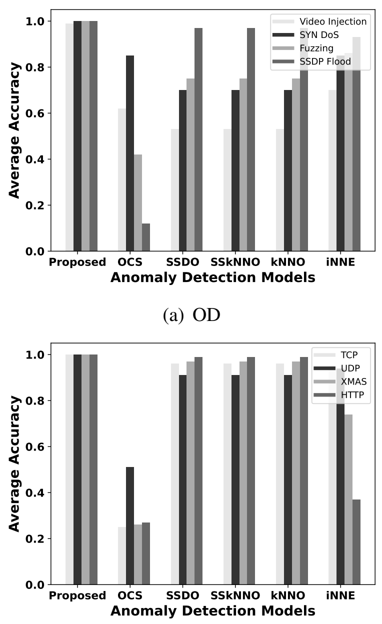
(b) UDP

(c) XMAS

Fig. 8. Comparison (average accuracy) between the proposed approach and others approaches: transfer learning across devices with TCP, UDP, and XMAS Flood Attacks in (a), (b), and (c), respectively.

We also repeat the above experiment using “SD” as the training data set and “OD” for testing. We collected data on Video Injection, SSL Renegotiation, ARP MitM, and Mirai attacks and used 100 random traffic packets from “SD” for the training phase. Next, we randomly selected 500 traffic data points from “OD” for testing. We computed the average accuracy for ten runs.

The results from the two experiments are presented in Figure 9 (a) and (b). We observe that, SSDO, SSkNNO, and kNNO methods have much better performance in anomalous traffic detection compared to OCS, and iNNE methods in the both experiments. In case of detecting the traffic packets affected by HTTP and SSDP Flood attacks, SSDO, SSkNNO, and kNNO methods show the same performance as ours. However, our approach outperforms SSDO, SSkNNO, kNNO, OCS, and iNNE methods in detecting attacks like Video Injection, SYN DoS, Fuzzing, TCP, UDP, and XMAS in terms



(a) OD

(b) SD

Fig. 9. Comparison (average accuracy) between the proposed approach and others approaches for unseen anomaly inference: (a) OD dataset for training and (b) SD dataset for training.

of average accuracy.

Comparison with Additional Unsupervised Approaches: In this set of experiments, we compared our approach with a larger set of state-of-the-art unsupervised anomaly detection methods that do not use transfer learning. In the first experiment, we leveraged the first dataset, where traffic packets are affected by flooding attacks: TCP, UDP, XMAS, and HTTP. For each of these attacks, we randomly selected 500 normal traffic packets to train all models, and we selected 1000 abnormal traffic packets for testing. For each attack, we ran the experiment five times and computed the average accuracy for all the cases. From the results, we observe that the OCS and OCS-SGD methods show better accuracy values than the ISOF, LOF, and RC unsupervised approaches in detecting traffic packets affected by flooding attacks. Our approach outperforms the OCS and OCS-SGD methods in terms of accuracy in all cases. In Figure 10(a), we demonstrate that our proposed approach significantly outperforms unsupervised anomaly detection models in each case.

In the second experiment, we considered four fold cross-validation for the transfer learning scenario across the attacks (as described earlier). For our first dataset "OD", we consider four flood attacks: TCP, UDP, XMAS, and HTTP. We focus on scenarios where the training data involves three attacks, and the testing data is the fourth attack. We repeated the experiment ten times, and computed the average accuracy. Figure 10(b) illustrates that the proposed approach significantly outperformed other unsupervised anomaly detection models in each case. Similarly, we used the second dataset "SD". Specifically, we leveraged the traffic data related to the 'SYN DoS' [25], 'OS Scan' [25], 'Fuzzing' [25], 'Mirai' [25] attacks to implement four fold cross validation experiment as mentioned earlier. For all models, we randomly selected 500 traffic data points for training and 1000 traffic data points for testing. For each case, we repeated the experiments five times, and computed the average accuracy. In Figure 10(c), we observe that our approach outperforms unsupervised models in terms of accuracy. We see that for both "OS Scan", and "Fuzzing" attacks, LOF has the accuracy values 0.99. On the other hand, for both "OS Scan", and "Fuzzing" attacks the accuracy values of our approach are 1.0. Although LOF performs very well in detecting anomalous traffic, our approach is more accurate in detecting anomalous traffic and outperforms LOF. In case of "Mirai" attacks, the accuracy value of RC method is 0.98 whereas our approach's accuracy value is 1.0. Therefore, our approach is more effective than other state-of-the-art anomaly detection methods for anomaly traffic detection.

Comparison with Supervised Approaches: In addition to these semi-supervised and unsupervised baselines, we have compared our approach with supervised models [28] such as SVM, KNN, Logistic Regression, Decision Tree, and Random Forest. We leveraged the IoT-23 data for this experiment. We have used random 160 network data points for training and 1500 network data points as the test data for all methods. We repeated the experiment ten times and computed the average accuracy. The results in Table II demonstrate that our unsupervised approach outperforms the supervised models in terms of accuracy. Specifically, the average accuracy of the proposed approach is 0.96, whereas, the average accuracy values of SVM, KNN, Logistic Regression, Decision Tree, and Random Forest are 0.86, 0.92, 0.91, 0.90, and 0.90, respectively. **Data and code along with additional analysis of our approach are available at:**https://anonymous.4open.science/r/anonymous12-F157/README.md

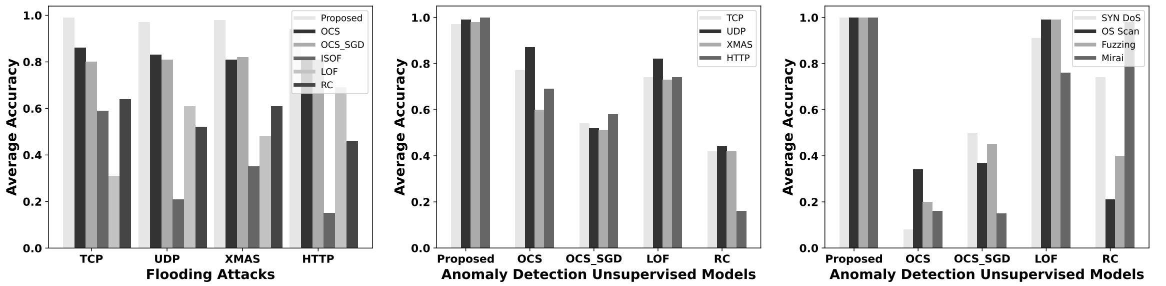
TABLE II
COMPARISON BETWEEN THE PROPOSED APPROACH AND SUPERVISED MODELS.

Methods	SVM	KNN	Logistic Regression	Decision Tree	Random Forest	Proposed Approach
Accuracy	0.86	0.92	0.91	0.90	0.90	**0.96**

.

IV. CONCLUSION

In this paper, we present a novel transfer-learning-based anomaly detection approach in which we accurately extract normal traffic data information from source data and detect unseen anomaly traffic in target data. Our approach efficiently leverages unsupervised algorithms, local outlier factors, and one-class SVM through the proposed transfer-learning approach. Our extensive evaluation shows the efficacy of our approach. In future, we will extend our research to classify the unknown attacks and to generate the human-interpretable explanations to justify the classification.



(a) Prediction Accuracy OD

(b) Transfer Learning Across Attacks on OD

(c) Transfer Learning Across Attacks on SD

Fig. 10. Comparison (average accuracy) between proposed approach and unsupervised state-of-the-art approaches: (a): prediction accuracy (without transfer learning), (b): transfer learning across attacks on first dataset, (c) transfer learning across attacks on second dataset.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[3] D. Das, U. Sharma, and D. Bhattacharyya, “Detection of http flooding attacks in multiple scenarios,” in *Proceedings of the 2011 international conference on communication, computing & security*, 2011, pp. 517–522.

[4] H. Wang, D. Zhang, and K. G. Shin, “Detecting syn flooding attacks,” in *Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies*, vol. 3. IEEE, 2002, pp. 1530–1539.

[5] M. Amer, M. Goldstein, and S. Abdennadher, “Enhancing one-class support vector machines for unsupervised anomaly detection,” in *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, 2013, pp. 8–15.

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[7] Y. Hou, S. G. Teo, Z. Chen, M. Wu, C.-K. Kwoh, and T. Truong-Huu, “Handling labeled data insufficiency: Semi-supervised learning with self-training mixup decision tree for classification of network attacking traffic,” *IEEE Transactions on Dependable and Secure Computing*, 2022.

[8] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *ICCV*. IEEE Computer Society, 2017, pp. 3449–3457.

[9] L. M. Manevitz and M. Yousef, “One-class svms for document classification,” *Journal of machine Learning research*, vol. 2, no. Dec, pp. 139–154, 2001.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[11] Z. Cheng, C. Zou, and J. Dong, “Outlier detection using isolation forest and local outlier factor,” in *Proceedings of the conference on research in adaptive and convergent systems*, 2019, pp. 161–168.

[12] C.-S. Shieh, T.-T. Nguyen, C.-Y. Chen, and M.-F. Horng, “Detection of unknown ddos attack using reconstruct error and one-class svm featuring stochastic gradient descent,” *Mathematics*, vol. 11, no. 1, p. 108, 2022.

[13] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, “Isolation-based anomaly detection using nearest-neighbor ensembles,” *Computational Intelligence*, vol. 34, no. 4, pp. 968–998, 2018.

[14] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[15] N. Patil, S. Menon, D. Das, and M. Pecht, “Anomaly detection of non punch through insulated gate bipolar transistors (igbt) by robust covariance estimation techniques,” in *2010 2nd International Conference on Reliability, Safety and Hazard-Risk-Based Technologies and Physics-of-Failure Methods (ICRESH)*. IEEE, 2010, pp. 68–72.

[16] V. Vincent, M. Wannes, and D. Jesse, “Transfer learning for anomaly detection through localized and unsupervised instance selection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 6054–6061.

[17] H. Nori, S. Jenkins, P. Koch, and R. Caruana, “Interpretml: A unified framework for machine learning interpretability,” *arXiv preprint arXiv:1909.09223*, 2019.

[18] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[19] S. D. Bay, “Combining Nearest Neighbor Classifiers through Multiple Feature Subsets,” in *Proceedings of the Fifteenth International Conference on Machine Learning*. Madison, WI: Morgan Kaufmann, 1998, pp. 37–45.

[20] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Baumer, and J. Davis, “Semi-supervised anomaly detection with an application to water analytics.” in *ICDM*, vol. 2018, 2018, pp. 527–536.

[21] L. Stradiotti, L. Perini, and J. Davis, “Semi-supervised isolation forest for anomaly detection,” in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 2024, pp. 670–678.

[22] J. Beale, A. Orebaugh, and G. Ramirez, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[23] A. Bijalwan, M. Wazid, E. S. Pilli, and R. C. Joshi, “Forensics of random-udp flooding attacks,” *Journal of Networks*, vol. 10, no. 5, p. 287, 2015.

[24] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, “Analysis of ddos-capable iot malwares,” in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017, pp. 807–816.

[25] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: an ensemble of autoencoders for online network intrusion detection,” *arXiv preprint arXiv:1802.09089*, 2018.

[26] A. Parmisano, S. Garcia, and M. Erquiaga, “A labeled dataset with malicious and benign iot network traffic,” *Stratosphere Laboratory: Praha, Czech Republic*, 2020.

[27] A. Takanen, J. D. Demott, C. Miller, and A. Kettunen, *Fuzzing for software security testing and quality assurance*. Artech House, 2018.

[28] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Ieee, 2016, pp. 1310–1315.