applied sciences

MDPI

Article

Towards Zero-Shot Flow-Based Cyber-Security Anomaly Detection Framework

Mikołaj Komisarek [1], Rafał Kozik [1,2], Marek Pawlicki [1,2,*] and Michał Choraś [1,2]

[1] ITTI Sp. z o.o., 61-612 Poznań, Poland
[2] Institute of Telecommunications and Computer Science, Bydgoszcz University of Science and Technology, 85-796 Bydgoszcz, Poland
* Correspondence: marek.pawlicki@itti.com.pl

Abstract: Network flow-based cyber anomaly detection is a difficult and complex task. Although several approaches to tackling this problem have been suggested, many research topics remain open. One of these concerns the problem of model transferability. There is a limited number of papers which tackle transfer learning in the context of flow-based network anomaly detection, and the proposed approaches are mostly evaluated on outdated datasets. The majority of solutions employ various sophisticated approaches, where different architectures of shallow and deep machine learning are leveraged. Analysis and experimentation show that different solutions achieve remarkable performance in a single domain, but transferring the performance to another domain is tedious and results in serious deterioration in prediction quality. In this paper, an innovative approach is proposed which adapts sketchy data structures to extract generic and universal features and leverages the principles of domain adaptation to improve classification quality in zero- and few-shot scenarios. The proposed approach achieves an F1 score of 0.99 compared to an F1 score of 0.97 achieved by the best-performing related methods.

Keywords: transfer learning; feature extraction; anomaly detection

Citation: Komisarek, M.; Kozik, R.; Pawlicki, M.; Choraś, M. Towards Zero-Shot Flow-Based Cyber-Security Anomaly Detection Framework. *Appl. Sci.* **2022**, *12*, 9636. https://doi.org/10.3390/app12199636

Academic Editor: Yu-Dong Zhang

Received: 10 August 2022
Accepted: 23 September 2022
Published: 26 September 2022

1. Introduction

The problem of transfer learning in the area of cyber-security is an important issue from the point of view of artificial intelligence. Currently, researchers are struggling with the issue of access to labelled data [1]. The labelling of new datasets is a time-consuming and problematic process [2]. There are several public datasets that can be used to train customized cyberattacks and anomaly detection algorithms [3]. However, experiments show that a model trained on network traffic from another network does not achieve satisfactory classification quality. This is to a large extent due to differences between the analysed networks, which result from the use of other services, the number of elements in the network, the way the network is used, and other factors [1]. To the best of the authors' knowledge, there is a limited number of papers which tackle flow-based network anomaly detection in the context of transfer learning, and these are mostly used on outdated datasets. The ambition of this paper is to fill this gap by proposing a new domain adaptation approach for anomaly detection. The main contributions of the paper can be summarized as follows:

- sketchy data structures are adapted for extracting generic and universal features and are compared with approaches described in the literature,
- the principles of domain adaptation are leveraged to improve classification quality in zero and few-shot scenarios,
- recently published and realistic datasets are used to compare the proposed approach under different scenarios with respect to anomaly detection.

This paper is structured as follows: In Section 2, related studies are described. Section 3 details the proposed method. Section 4 describes the experimental setup, metrics and methodology. Section 5 describes the results. The paper ends with conclusions.

2. Related Work

The problem of transfer learning in the cyber-security domain is a difficult issue that stems from the diversity of tools used by adversaries, privacy-related restriction (i.e., constraints in sharing data with the community), and the complexity of systems that face the dynamically evolving landscape of cyber-threats.

However, despite these obstacles, significant effort has been invested by researchers to define mechanisms that would allow system administrators to evolve detection systems from data-driven to knowledge-driven solutions. In this regard, there is an urgent need for a solution that would allow extraction of useful patterns that can be used in recognising unknown cyber-attacks.

The authors of [2] enhanced transfer learning with clustering. The approach, named CeHTL, was able to uncover how a new attack was related to already known attacks. However, in contrast to the approach presented in this paper, the research of [2] is based mainly on the NSL-KDD dataset, which contains traffic collected in 1999 [4], and self-generated, synthetic datasets.

In [5], the authors introduced a transfer-learning-based method to tackle the imbalanced data issue in cyber-security. Although this approach achieved promisingly high F1-scores, the authors noted that their method is impractical if the minority class samples are rare.

Another interesting approach was presented in [6]. The authors adapted a semi-supervized learning method utilising a recurrent variational autoencoder (RVAE). The method aims at capturing sequential characteristics of botnet activities. Similar to the approach presented in this paper, the method uses network flow characteristics to capture various behavioural patterns. However, this method results in a relatively high number of false alarms.

The authors of [7] proposed a network intrusion detection (NIDS) framework featuring a deep neural network. The network was established on a pretrained VGG-16 architecture. Using a transfer-learning-for-network-intrusion-detection (TL-NID) framework, in the first step, the features were extracted with the use of the VGG-16. The network was pre-trained on an ImageNet dataset. In the second step, a deep neural network was applied to the extracted features for classification. The approach was tested on the dated benchmark NSL-KDD. To enable the approach to work, the samples from the NSL-KDD dataset were transformed to images conforming with the VGG-16 input shape.

In [8], the authors proposed a multi-source transfer learning intrusion detection system (IDS) to work with encrypted data. The method enabled successful transfer of knowledge from encrypted models in multiple source domains to the target domain, with an accuracy exceeding 93%. The authors tested the use of the proposed E-XGBoost transfer learning method on the CTU-13 dataset.

The authors of [9] pointed out that the direct utilization of classes coming from a different network is not sufficiently accurate to detect anomalous behaviours in a new, target network. To counter this, the authors put forward a method to transfer knowledge between networks to eliminate the need for training samples in the target network. The method is based on manifold alignment-leveraging domain-adaptation manifold alignment (DAMA) from [10] to unify source and target feature spaces, along with adaptation regularization for transfer learning using squared loss from [11]. The proposed method successfully transferred knowledge from the NSL-KDD source dataset to the target domain based on the Kyoto2006 dataset, with accuracy and recall exceeding 90%.

Zero-shot learning was proposed to address the issue of detecting unknown attacks in [12]. The authors treated part of the feature vector as a semantic description of the

attacks found in the benchmark used. Part of the benchmark was treated as a corpus to train a Word2VEC.

A self-taught learning approach to IDS was introduced in [13]. The method relies on feature extraction using adaptive self-taught learning. This is achieved with the use of a sparse autoencoder. The source domain also relies on time-series data, but is not cybersecurity-related. The features extracted from the target domain data by the sparce autoencoder trained on the source domain data were used in conjunction with the original feature vector to train and test a deep neural network and a deep belief network, and applied using non-linear principal component analysis.

In [14], an entire deep learning pipeline with a set of specific improvements for better detection results using the neural networks employed in IDS was presented.

The authors of [15] addressed the lack of historical data on intrusion detection by introducing zero-shot learning, which can help with anomaly detection in circumstances of insufficient data samples by replacing the necessary knowledge with semantic estimation. The approach was tested for scenarios of insider threat where historical data were unavailable. The existing IDS was augmented with descriptions of user positions, roles and project assignments, which were incorporated through graph embeddings.

In [16], the authors utilized raw packet capture (PCAP) files from the BoT-IoT dataset, and then used the Argus tool to extract header-field-information-based features (rather than flow-based aggregations). The authors used the embedding layers from a multi-class classification model as feature extractors for a binary classification model. The authors of [17] used a ResNet50 pre-trained model as a feature extractor and fine-tuned it by training a fully connected dense layer connected to its outputs. The model was trained on the MalImg benchmark and exceeded 99% accuracy. A similar approach was used in [18].

Network intrusion detection using deep learning employing domain adaptation was explored in [19]. The authors addressed the problem of the scarcity of data by use of domain adaptation techniques and transferring knowledge from a labelled NIDS dataset, with the feature spaces remaining the same between the source domain and the target domain. The domain adaptation was performed using generative adversarial networks (GANs), where the generator was trained to perform domain-invariant mapping of both the source and the target domains, which was then used as input for the classifier.

A formalized method to set the decision threshold for transfer-learning-based anomaly detection was described in [20]. The authors noted that deep learning can often be used for feature extraction, with the extracted features then subject to comparison with a model of normality. The report emphasized that setting the threshold used for the comparison properly enables the approach to outperform other approaches.

In [21], a zero-shot intrusion detection method leveraging a regression model was introduced. The classification was performed inductively by regression fitting for each category and calculating the decision threshold. The model was able to detect unknown attack types.

In [22], a taxonomy of transfer learning techniques was presented, with the division into classes depending on what kind of data are available, and whether labelled data are to be found in the source, in the target domain, or if there is no labelled data at all. The authors provide descriptions of the types of algorithms helpful in the circumstances described.

In [23], the authors proposed a hybrid contrastive model (HCM) to perform identity-level, along with image-level, contrastive learning for unsupervized reidentification, which exploits feature similarity between hard sample couples.

The main conclusion to be drawn from the above literature analysis is that there is a limited number of papers which tackle flow-based network anomaly detection in the context of transfer learning. The existing methods often use outdated datasets (e.g., NSL-KDD) or formats (e.g., raw PCAP files) that are difficult to obtain at a larger scale (e.g., for privacy reasons). Therefore, our ambition in this paper is to fill this gap by proposing a new domain adaptation approach for anomaly detection. Existing flow-based techniques often rely on a raw network flow format [24], which may narrow the analysis context.

Therefore, we propose sketchy feature vectors (SFV), which enable us to capture additional characteristics representing the behaviour of network elements.

3. Proposed Method

The key goal of the proposed solution, as shown in Figure 1, is to enable the system operator to avoid extensive model training when (i) the network environment changes, or (ii) there is a need to move detectors from one network to another. The proposed solution comprises the following building blocks:

- Network probe, which captures network traffic in the form of network flows,
- Sketchy feature vectors (SFV) extraction, which calculates feature vectors over a predefined time window,
- Anomaly detection and threat identification, which is responsible for detecting anomalies in the observed traffic and categorizing them as a known threat,
- Domain adaptation module, which is intended to bring the traffic coming from a different network onto a feature space where the anomaly detection and threat classification were trained,
- Dashboard, which is intended to visualize various traffic characteristics for the identified anomalies and threats.

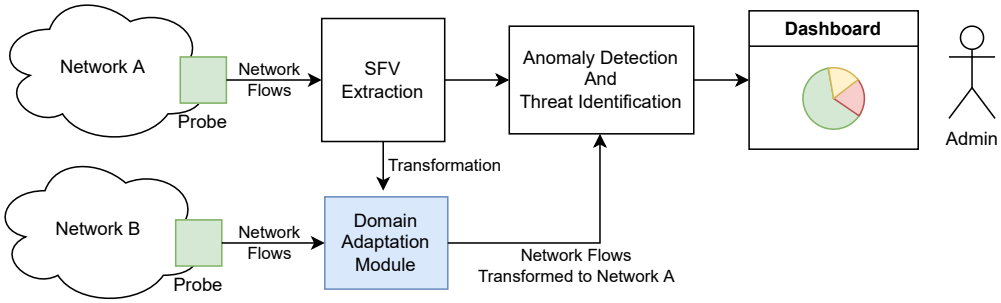

Figure 1. The architecture of the proposed solution.

3.1. Flow-Based Data Acquisition

In this approach, the gathered network data is of the communication flow type [25]. It may come from a broad range of devices, e.g., switches, routers and hosts; it features the properties and statistics pertaining to a network in an aggregated manner. As far as the architecture of the flow-enabled devices is concerned, collectors are the elements where the collected traffic is sent. Subsequently, it is stored and kept there for further analysis, which is usually performed as part of auditing activities by network administrators. In a single flow, the following characteristics are collected:

- the number of incoming and outgoing bytes
- IP addresses partaking in the communication
- utilized source and destination ports
- utilized type of protocol (e.g., transmission control protocol (TCP) or user datagram protocol (UDP))

Network node anomalous behaviour patterns (parameter changes in network flows) must be identifiable, and this kind of data ought to enable identification, as the patterns can be indicative of malware infection. Thus, a network administrator is able to utilize them in order to recognize an adversary.

3.2. SFV—Sketchy Feature Vectors Extraction

In the following approach, before the statistical properties are determined, the group flows in question are gathered in relation to a specific IP address, in so-called time windows, i.e., time spans which are relatively short and of fixed length.

In a preceding study by the authors ([26]), it was highlighted that a single flow has multiple characteristics that define the two-way communication (such as the number of flows or destination IP). It is possible to compute different statistical properties, such as mean, median value, and min/max values, for all the characteristics.

When determining how many flows, as well as inbound and outbound packets, there are, the specific counting or identification of the elements that are the most frequent in the datastreams (such as destination ports) is a demanding task. To overcome this, a straightforward solution is the maintenance of a dynamic list. Using this approach, if an unknown element is fetched from the flow, the whole list must be examined to confirm if the element is present. In the event that it is not there, the element has to be added to the list, which in turn is resized. In addition, if there are various computational processes running simultaneously, and there is a need to merge their results, it adds a further level of complexity.

In [27], a data structure class is discussed, called probabilistic (or sketchy). This class of data structure is able to describe exceptionally large sets, with sub-logarithmic/constant space complexity. In this way, it is not necessary for the data-processing system to be scaled up, even if, for example, it has to transition from analysing thousands to billions of records.

Probabilistic data structures utilize a number of distinct data-compressing mechanisms; these might result in the structures containing inaccurate information.

Despite these inaccuracies, the detection part should not be influenced to a significant degree. This assumption results from the fact that, to a certain degree, the classifiers are able to deal with this kind of change and make the correct decision. It should be recalled that, in this instance, the changes in question are in the range of 1–2% for a feature constituting the vector.

Probabilistic data structures offer a number of potential benefits. Firstly, their size increases (often much) less slowly in relation to the growing amount of input data. In addition, making the trade-off between the accuracy of prediction and the size of the data structure also proves to be feasible.

These structures are suitable for processing network traffic streaming data, as every element in the stream requires swift analysis and updating of the data structure, by summarising a number of properties (such as the number of distinct IP addresses or the service which is used most frequently). The ability of probabilistic data structures to be merged is shown to be feasible. In other words, the stream can be divided into two pieces and the calculations performed separately, and this procedure will produce the same outcome as if performed over the whole (original) stream. Consequently, probabilistic structures are highly parallelizable and, thus, in compliance with distributed computing platforms, such as Hadoop, Spark, Druid, and others.

Data structures such as a hash table can be used to compute the most frequent destination port or destination host that originate from a specific IP address, or a combination of both. When doing so, the new item goes into the hash table, with the counter being set to 0 for the item. Where an entry in the table already exists, the counter is incremented. However, with a vast quantity of input data, such an approach is prone to becoming unattainable. The reason for this is that the hash table expands along with increase in the amount of input data; eventually, there is not sufficient RAM available to proceed. The collisions in the hash table are treated as a linked list. In other words, in cases where a new item is hashed to an already taken bucket, it is appended/linked after the existing one. In this way, as the list expands, it takes more and more time to access the items in the hash table. Moreover, allocating memory for a new element in a dynamic way also consumes time.

To overcome this problem, in this research, two types of probabilistic data structures were used, namely Count-Min (CM) and HyperLogLog (HLL), for frequency and cardinality estimation, respectively. The Count-Min (CM) data structure enables the counting of items that are of a different type, e.g., how many times a specific IP address has established a TCP

connection. On the other hand, HyperLogLog (HLL) belongs to a family of algorithms that aim at estimating the cardinality of a dataset (e.g., the number of distinct destination ports).

3.3. Domain Adaptation

The general concept of domain adaptation is illustrated in Figure 2. First, let us denote X as a feature space and x as a feature vector. In particular, a vector having d attributes is denoted as $x = [x^{(1)}, x^{(2)}, \ldots, x^{(d)}]$, and $x \in X = X^{(1)} \times X^{(2)} \times \cdots \times X^{(d)}$.

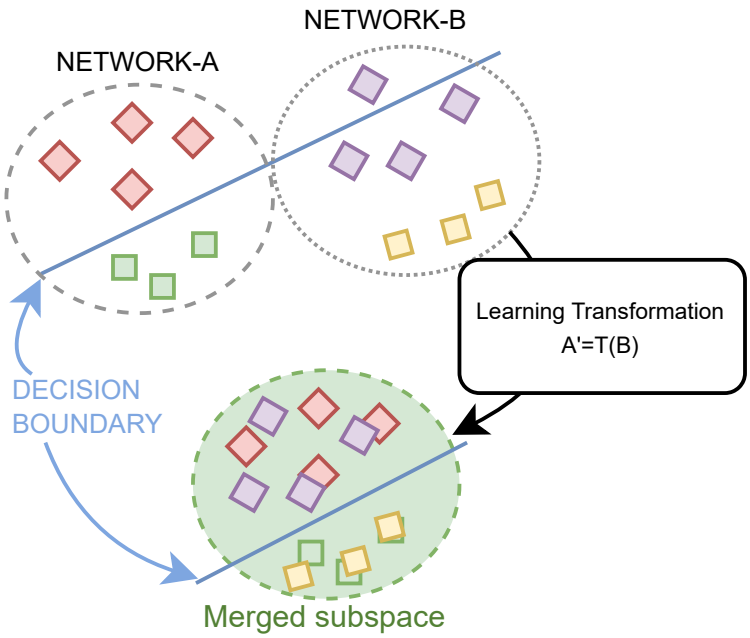

Figure 2. Model transfer—general overview of domain adaptation for traffic recorded for Network B to feature space of Network A.

The idea behind the proposed domain adaptation method is to project the feature vectors $x_i^{(d)}$ recorded for Network B, where $i \in [1, m]$, onto a feature space of Network A, for which the original classifier has been trained. In other words, this approach resembles the batch normalization concept widely used when training artificial neural networks. In principle, after applying the inner-bracket part of Equation (1), we obtain a zero mean and unit variance matrix of feature vectors. Next, we transform this matrix (using the outer part of Equation (1)), so that the feature vectors become aligned with the source domain, where the original classifier has been trained. These two steps are implemented by sequentially executing transposition and scaling operations according to Equation (1), where $T_{A,B}$ and $S_{A,B}$ indicate transposition and scaling applied for network A or B, and x represents the collection of feature vectors.

$$T_{B \to A}(x) = \left[(x - T_B) S_B^{-1} \right] S_A + T_A \tag{1}$$

In this way, the T and S can be estimated separately for different networks. Here, T and S are considered as transformations implementing the standardization process, so that the mean becomes zero and the standard deviation becomes one. More precisely, $T_{\{A,B\}}$ and $S_{\{A,B\}}$ are calculated using Equations (2) and (3).

$$T_{\{A,B\}} = 1_m \mu_{\{A,B\}}^T \tag{2}$$

$$S_{\{A,B\}} = diag(\sigma_{\{A,B\}}) \tag{3}$$

In both formulas, μ and σ are calculated as the classical mean and standard deviation using Formulas (4) and (5), respectively. Moreover, 1_m is $m \times 1$ vector of ones.

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{4}$$

$$\sigma = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2 \tag{5}$$

3.4. Anomaly Detection and Threat Identification

Anomaly detection and threat identification involves a two-stage cascade, which is represented in Figure 3. The approach utilises several random forest classification models. The first one in the cascade is responsible for binary classification, which indicates whether a feature vector is considered normal or anomalous. The second part of the cascade is responsible for threat identification. It is triggered only if an anomaly is detected. This approach enables achievement of a modular architecture, where new threat detection models can be added at anytime, without retraining the whole system from the beginning. Moreover, it may also happen that an anomaly alert will be triggered when none of the threat identification modules can provide an unequivocal answer.

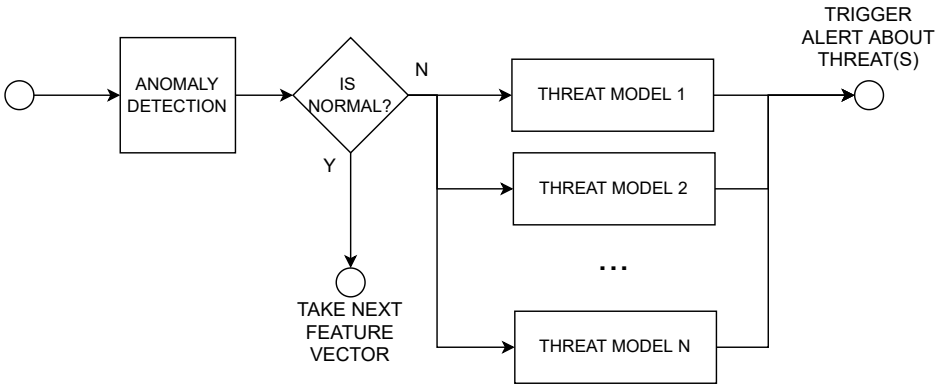

Figure 3. The model cascade used for anomaly detection and threat identification (there are two steps of classification in the proposed approach: anomaly detection and threat identification).

4. Experimental Setup

4.1. Experiments

The aim of the experiments was to evaluate the effectiveness of the proposed solution with respect to anomaly detection and threat identification, as well as to assess the potential for transferring the trained models between different networks. For this reason, two datasets were selected, which were similar in terms of the recorded cyber threats (e.g., both datasets consisted of similar attacks, such as network scan or DDoS). However, both datasets differed in terms of network size, the volume of recorded traffic, etc.

4.2. Datasets Used for Evaluation

In the experiments, two datasets were employed, namely, IoT-23 and SIMARGL2021.

IoT-23 [28] is a dataset containing network traffic sourced from the IoT (Internet of Things) devices, containing three captures for benign IoT device traffic and twenty malware captures. It was circulated in January 2020 for the first time; the captures come from 2018 and 2019. The particular IoT network traffic was captured in the Stratosphere Laboratory, AIC group, FEL, CTU University, in the Czech Republic. The main objective was to provide a comprehensive dataset of real and labelled IoT malware infections and IoT benign traffic for researchers to develop machine learning algorithms. Both the dataset and the research associated with it were sponsored by Avast Software, Prague.

SIMARGL2021 [29] is a dataset assembled from a real-world, academic network, from which real-life traffic was gathered after having carried out a variety of attacks. The format

selected for the network data schema is Netflow v9. It encompasses 44 specific features; each frame is labelled.

4.3. Metrics Used for Evaluation

Prior to applying a number of different machine learning algorithms, the raw network flows were handled to obtain the sketchy feature vectors, according to the procedure described in the previous sections. The classification quality metrics were calculated according to the following procedure:

1. communication flows were aggregated into time windows (in this case, 3-min time windows were used).
2. for the given time windows, sketchy feature vectors were calculated.
3. within the ground-truth communication flows, labels were examined against those predicted; subsequently the TP, TN, FP and FN errors (true and false positives and negatives) were measured.
4. lastly, recall, precision, and F1-score metrics were calculated and reported.

4.4. Evaluation Methodology

The experiments were divided into three categories:

- First, the effectiveness of the proposed approach was evaluated separately on the IoT-23 and SIMARGL2021 datasets. A classical random split approach was used, where 70% of the data was used for training and the remaining 30% was used during testing. The recall, precision, and F1-score were measured for two cases, namely, anomaly detection and threat identification.
- Subsequently, the transferability capabilities of the proposed approach in a zero-shot manner were measured. The models were trained on the SIMARGL2021 dataset and evaluated on the IoT-23 dataset. The results for two cases were provided, namely, when the domain adaptation module was turned off and on. This enabled highlighting of the importance of domain adaptation for the proposed method.
- Finally, the transferability capabilities were tested using a varying number of samples drawn from the other domain. Specifically, the models were trained on the SIMARGL2021 dataset with N additional samples from IoT-23, and evaluated using the models for the remaining part of the IoT-23 dataset.

5. Results

The analyses of the results have been divided into effectiveness comparison (Section 5.1), zero-shot scenario evaluation (Section 5.2), and few-shot scenario evaluation (Section 5.3).

5.1. Effectiveness Comparison

In this section, the evaluation results for the anomaly and threat identification obtained for IoT-23 and SIMARGL2021 datasets are presented separately. It can be seen (see Tables 1 and 2) that the proposed approach achieved very good results for both datasets (the F1-score metric was higher than 0.9 for all classification tasks).

Table 1. Detection effectiveness of the proposed approach (IoT-23 dataset).

Class	Recall	Precision	F1-Score
Benign	0.9980	0.9968	0.9974
Anomaly	0.9974	0.9984	0.9979
CNC	1.0000	0.9957	0.9978
DDOS	1.0000	0.8814	0.9369
Okiru	1.0000	0.9990	0.9995
Torii	1.0000	1.0000	1.0000
PortScan	0.9972	0.9954	0.9963

In the experiments, two classification scenarios were considered. One was focused on anomaly detection, while the other was related to thread identification. In the first case (anomaly detection), all the samples that indicated any kind of malicious behaviour were given the label 'anomalous', and genuine traffic samples were indicated as 'normal'. Having prepared the data, the ML-based model was trained according to the procedure presented in the previous section. For the second case (threat identification), the ML models were trained in a one-to-many fashion. In other words, a dedicated classifier was trained for each threat (cyber-attack).

Table 2. Detection effectiveness of the proposed approach (SIMARGL2021 dataset).

Class	Recall	Precision	F1-Score
Benign	1.0000	0.9999	1.0000
Anomaly	0.9716	0.9884	0.9799
RUDY	0.9941	0.9883	0.9912
Slowloris	0.9947	1.0000	0.9973
FIN Scan	0.9710	1.0000	0.9853
NULL Scan	0.9761	1.0000	0.9879
UDP Scan	0.9907	1.0000	0.9953
XMAS Scan	0.9552	1.0000	0.9771

In both experiments, there were cyber threats that were conceptually similar in terms of techniques used by the adversaries. In particular, the malware included in the IoT-23 datasets conveyed network reconnaissance, which relied on network scanning, which, in turn, was included in the SIMARGL2021 dataset. The RUDY and Slowloris traces from the SIMARGL2021 dataset should resemble the samples that were indicated as DDoS.

However, both datasets were significantly different in terms of the tools used to implement the attack, as well as the technical means to record the network traces (e.g., to record SIMARGL dataset, nProbe was used, while IoT-23 utilized Zeek/Bro firewall).

5.2. Zero-Shot Scenario

In this section, the results obtained for the zero-shot approach scenario are presented. The values of the evaluation metrics are presented in Table 3. Two cases were considered. The first was a model that was trained on the SIMARGL2021 dataset and directly used on the IoT-23 dataset during the evaluation process. The second utilized the "domain adaption" mechanism described in the previous sections. In Table 3, significant differences in the obtained results are marked. The data demonstrate that the mechanism is useful and helps to improve the results obtained.

Table 3. Zero-shot scenario. Model trained on SIMARGL dataset and tested on IoT-23, (with and without domain adaptation).

Scenario	Class	Recall	Precision	F1-Score
Without Domain Adaption	Benign	0.69481	0.6295	0.66055
	Anomaly	0.66133	0.7235	0.69102
With Domain Adaption	Benign	0.6823	**0.8450**	**0.7550**
	Anomaly	**0.7965**	0.6065	0.6886

5.3. Few-Shot Scenario

In this section, the results obtained for the few-shot approach scenario are presented. The values of the evaluation metrics are shown in Table 4. Three cases were considered. For each of the cases, the SIMARGL2021 dataset was enriched with additional N samples from the IoT-23 dataset. The number was gradually increased, starting from 100 traces, which

represented a few minutes of recorded network flow samples. As shown in Table 4, as few as 100 additional samples produced a significant boost in terms of accuracy.

Table 4. Few-shot scenario. Model trained on the SIMARGL dataset (with N additional samples from IoT-23), and tested on IoT-23.

Samples	Class	Recall	Precision	F1-Score
100	Benign	0.9122	0.9500	**0.9307**
	Anomaly	0.9478	0.9085	**0.9278**
500	Benign	0.9497	0.9730	0.9612
	Anomaly	0.9723	0.9485	0.9603
1000	Benign	0.9875	0.9880	0.9878
	Anomaly	0.9880	0.9875	0.9878

5.4. Comparison of Results with Other Methods

In this section, we compare the proposed detection method with other approaches described in the literature. As SIMARGL2021 is a relatively new dataset, we have focused the comparison on the IoT-23 dataset. The results are presented in Table 5. For brevity, we selected the average F1-score as a basis for comparison of the different methods. This is because the F1-score is always reported by researchers and is a better metric for performance evaluation when imbalanced data is considered [30]. In the comparison, we have included both classical (shallow) (e.g., [24]) and deep learning approaches (e.g., [31]).

Table 5. Comparison of methods.

Method	Average F1-Score
Proposed method	**0.99**
Adversarial Autoencoders + KNN [31]	0.97
BiGAN + KNN [31]	0.97
AdaBoost [24]	0.83
SVM [24]	0.59

6. Conclusions and Future Work

In this paper, an innovative approach is proposed, which adapts sketchy data structures for extracting generic and universal features, and leverages the principles of domain adaptation to improve classification quality in zero- and few-shot scenarios. The experiments and the reported results enable us to conclude that the proposed mechanism can be successfully used in difficult anomaly detection and threat identification scenarios. Although the datasets utilized during the experiments were essentially different in terms of tools (e.g., used for recording the datasets traffic traces), and the techniques used by the adversaries to implement the attack, it was possible to extract common attack patterns that can be successfully used to detect abnormal behaviour of network elements. Although the presented "domain adoption" mechanism has already been demonstrated to be a useful tool, the authors plan to further enhance it using additional information sources. In particular, the authors believe that by analysing multi-domain network-flow-based knowledge transfer, it will be possible to extract general patterns that can help form better model decision boundaries when transferring knowledge across various domains.

Author Contributions: Conceptualization, methodology, software, validation, investigation, writing, M.K. and R.K.; formal analysis, review and editing, project administration, M.P. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded under the APPRAISE (fAcilitating Public Private secuRity operAtors to mitigate terrorIsm Scenarios against soft targEts) project, with the support of the European Commission and the Horizon 2020 Program, under Grant Agreement No. 101021981.

Data Availability Statement: The study was performed on open, public benchmark datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jung, I.; Lim, J.; Kim, H.K. PF-TL: Payload Feature-Based Transfer Learning for Dealing with the Lack of Training Data. *Electronics* **2021**, *10*, 1148. [CrossRef]
2. Zhao, J.; Shetty, S.; Pan, J.W.; Kamhoua, C.; Kwiat, K. Transfer learning for detecting unknown network attacks. *Eurasip J. Inf. Secur.* **2019**, *2019*, 1. [CrossRef]
3. Cremer, F.; Sheehan, B.; Fortmann, M.; Kia, A.N.; Mullins, M.; Murphy, F.; Materne, S. Cyber risk and cybersecurity: A systematic review of data availability. *Geneva Pap. Risk Insur.-Issues Pract.* **2022**, *47*, 698–736. [CrossRef] [PubMed]
4. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
5. Wang, H.; Liu, P. Tackling Imbalanced Data in Cybersecurity with Transfer Learning: A Case with ROP Payload Detection. *arXiv* **2021**, arXiv:2105.02996. https://doi.org/10.48550/ARXIV.2105.02996.
6. Kim, J.; Sim, A.; Kim, J.; Wu, K.; Hahm, J. Improving Botnet Detection with Recurrent Neural Network and Transfer Learning. *arXiv* **2021**, arXiv:2104.12602.
7. Masum, M.; Shahriar, H. Tl-nid: Deep neural network with transfer learning for network intrusion detection. In Proceedings of the 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 8–10 December 2020; pp. 1–7.
8. Xu, M.; Li, X.; Wang, Y.; Luo, B.; Guo, J. Privacy-preserving multisource transfer learning in intrusion detection system. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e3957. [CrossRef]
9. Taghiyarrenani, Z.; Fanian, A.; Mahdavi, E.; Mirzaei, A.; Farsi, H. Transfer learning based intrusion detection. In Proceedings of the 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 25–26 October 2018; pp. 92–97.
10. Wang, C.; Mahadevan, S. Heterogeneous domain adaptation using manifold alignment. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
11. Long, M.; Wang, J.; Ding, G.; Pan, S.J.; Philip, S.Y. Adaptation regularization: A general framework for transfer learning. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1076–1089. [CrossRef]
12. Zhang, Z.; Liu, Q.; Qiu, S.; Zhou, S.; Zhang, C. Unknown attack detection based on zero-shot learning. *IEEE Access* **2020**, *8*, 193981–193991. [CrossRef]
13. Qureshi, A.S.; Khan, A.; Shamim, N.; Durad, M.H. Intrusion detection using deep sparse auto-encoder and self-taught learning. *Neural Comput. Appl.* **2020**, *32*, 3135–3147. [CrossRef]
14. Pawlicki, M.; Kozik, R.; Choraś, M. A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. *Neurocomputing* **2022**, *500*, 1075–1087. [CrossRef]
15. Zerhoudi, S.; Granitzer, M.; Garchery, M. Improving intrusion detection systems using zero-shot recognition via graph embeddings. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 790–797.
16. Ge, M.; Syed, N.F.; Fu, X.; Baig, Z.; Robles-Kelly, A. Towards a deep learning-driven intrusion detection approach for Internet of Things. *Comput. Netw.* **2021**, *186*, 107784. [CrossRef]
17. Kumar, S. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in internet of things. *Future Gener. Comput. Syst.* **2021**, *125*, 334–351.
18. Mehedi, S.T.; Anwar, A.; Rahman, Z.; Ahmed, K.; Rafiqul, I. Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-based Approach. *IEEE Trans. Ind. Inform.* **2022**. [CrossRef]
19. Singla, A.; Bertino, E.; Verma, D. Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, 1–5 June 2020; pp. 127–140.
20. Aburakhia, S.; Tayeh, T.; Myers, R.; Shami, A. A transfer learning framework for anomaly detection using model of normality. In Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 4–7 November 2020; pp. 0055–0061.
21. Zhang, X.; Gao, L.; Jiang, Y.; Yang, X.; Zheng, J.; Wang, H. A zero-shot intrusion detection method based on regression model. In Proceedings of the 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD), Suzhou, China, 21–22 September 2019; pp. 186–191.
22. Agarwal, N.; Sondhi, A.; Chopra, K.; Singh, G. Transfer learning: Survey and classification. In *Smart Innovations in Communication and Computational Sciences*; Springer: Singapore, 2021; pp. 145–155.

23. Si, T.; He, F.; Zhang, Z.; Duan, Y. Hybrid Contrastive Learning for Unsupervised Person Re-identification. *IEEE Trans. Multimed.* **2022**. [CrossRef]
24. Stoian, N. Machine Learning for Anomaly Detection in IoT Networks: Malware Analysis on the IoT-23 Data Set. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2020.
25. Choraś, M.; Pawlicki, M. Intrusion detection approach based on optimised artificial neural network. *Neurocomputing* **2021**, *452*, 705–715. [CrossRef]
26. Kozik, R.; Pawlicki, M.; Choraś, M. A new method of hybrid time window embedding with transformer-based traffic data classification in IoT-networked environment. *Pattern Anal. Appl.* **2021**, *24*, 1441–1449. [CrossRef]
27. Singh, A.; Garg, S.; Kaur, R.; Batra, S.; Kumar, N.; Zomaya, A.Y. Probabilistic data structures for big data analytics: A comprehensive review. *Knowl.-Based Syst.* **2020**, *188*, 104987. [CrossRef]
28. Garcia, S.; Parmisano, A.; Erquiaga, M.J. IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic. 2020. Available online: https://www.stratosphereips.org/datasets-iot23 (accessed on 22 September 2022). [CrossRef]
29. Mihailescu, M.E.; Mihai, D.; Carabas, M.; Komisarek, M.; Pawlicki, M.; Hołubowicz, W.; Kozik, R. The Proposition and Evaluation of the RoEduNet-SIMARGL2021 Network Intrusion Detection Dataset. *Sensors* **2021**, *21*, 4319. [CrossRef]
30. Wardhani, N.W.S.; Rochayani, M.Y.; Iriany, A.; Sulistyono, A.D.; Lestantyo, P. Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data. In Proceedings of the 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Tangerang, Indonesia, 23–24 October 2019; pp. 14–18. [CrossRef]
31. Abdalgawad, N.; Sajun, A.R.; Kaddoura, Y.; Zualkernan, I.; Aloul, F. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. *IEEE Access* **2021**, *10*, 6430–6441. [CrossRef]