International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Efficient Inductive Transfer Learning based Framework for Zero-Day Attack Detection

[1]Gunupusala Satyanarayana, [2]Kaila Shahu Chatrapathi
[1]Research Scholar, JNTU Hyderabad, India.
Email: snarayana.5813@gmail.com
[2]Professor, Department of CSE, JNTU Hyderabad, India.
Email: shahujntu@gmail.com

Abstract—An Intrusion Detection System (IDS) is a type of security domain that tracks and evaluates network connections or system operations to detect potential security breaches, unauthorized usage, and malicious activity within computer networks. Machine learning (ML) and deep learning (DL) algorithms provide better IDS based on the labelled dataset. However, due to a lack of labelled data, its effectiveness in detecting zero-day attacks is limited. Anomaly detection methods frequently produce high False Positive Rates (FPR). Transfer learning (TL) is a powerful technique in various domains, including intrusion detection systems (IDS). It also creates advanced classifiers using knowledge extracted from the related source domain(s) with little or no labelled data. This paper introduced zero-day attack detection (ZDAD) model by combining it with transfer learning that helps classify the attacks and non-attacks from the given dataset. Using the UNSW-NB15 dataset, the authors created a Transfer Learning-based prototype in this study. The goal was to unify the feature space for distinguishing unlabeled Generic samples representing zero-day attacks from regular instances using labelled DoS samples. The ZDAD performed admirably, achieving 99.24% accuracy and a low False Positive Rate (FPR) of 0.02%. This performance outperforms current state-of-the-art methods.

Keywords- Intrusion Detection Systems, Zero-day Attack, Transfer Learning, Target Domain, Source Domain

I. INTRODUCTION

IoT networks are fast integrating into diverse sectors like agriculture, healthcare, energy, transportation, and manufacturing, and they are interlinking billions of devices. Analysts estimate that the number of linked devices will reach 53 billion by the end of 2023. Nonetheless, this IoT device's extensive utilization has exposed IoT networks to a range of cyber threats. In 2022, sources [1] reported that bots, worms, and DDoS (Distributed Denial of Service) were prevalent assaults in networks of IoT. Furthermore, Kaspersky, an antivirus and security service provider, stated that the number of cyber-attacks targeting IoT networks doubled in 2021.These attacks pose significant threats to IoT systems, leading to detrimental consequences. One of the major challenges in securing IoT networks is that traditional security mechanisms, such as traditional Intrusion Detection Systems (IDS), prove to be excessively resource-intensive for IoT environments. Furthermore, a notable portion of IoT devices lack adequate security measures, as they are frequently manufactured without stringent security controls in place. As a result, a significant percentage of these devices inherently possess security vulnerabilities [2].

In recent years, significant research has been dedicated to enhancing IoT security through the application of ML and DL ("Deep Learning") approaches, particularly for IDS. At first, ML methods [3] were extensively utilized but faced difficulties like low detection rates and constrained feature engineering [4]. ML-based approaches also encountered challenges in identifying different forms of intrusions and threats, particularly unpredictable and unforeseen attacks. To address these limitations, DL techniques were embraced, enhancing the capabilities of ML-based solutions by detecting patterns that deviate from normal behavior. This advancement resulted in improved detection accuracy as well as decreased false positives [5, 6].

DL-based IDSs have proven to be effective in capturing intricate patterns for intrusion detection when trained on large labeled datasets. However, in IoT environments, obtaining such extensive labeled datasets for even known attacks or unknown (zero-day) attack families can be a challenge. The process of acquiring new training data in these networks is often expensive, time-consuming, or even non-existent. Additionally, when a new intrusion is found, DL models require complete retraining from scratch with the new data, which demands substantial calculating resources and time. Consequently, DL-based IDSs encounter difficulties in IoT networks due to the scarcity and imbalance of datasets, as well as the limited computing capabilities of IoT devices.

To overcome the challenges associated with limited and unbalanced datasets in detecting zero-day attacks, TL has developed as a promising solution [7]. A new development in ML called TL uses information from a source domain which is linked to target domain to enhance learning there. TL facilitates the creation of high-performance models for the target domain by utilizing the information from source domain. This approach has shown effectiveness in various domains such as NLP ("Natural Language Processing") [8] and CV ("Computer Vision") [9]. For instance, repurposing image classification models trained on one domain for a new, related domain can yield improved results compared to training the new dataset from scratch. TL has also been recently explored in the context of Intrusion Detection Systems (IDSs), demonstrating improvements in detecting

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

known attacks in data-scarce domains like IoT networks and enhancing the detection capabilities for zero-day attacks.

Research demonstrates that TL-based models achieve comparable performance to DL models even when trained with a small percentage (1 to 10%) of labeled training data. TL has proven to be a valuable approach for enhancing the performance of Intrusion Detection Systems (IDSs). It has been successfully employed to improve detection accuracy for new intrusions, expedite the process of training, and effectively detect zero-day attacks. By leveraging prior knowledge from a related source domain, TL enables IDSs to adapt and generalize to new and evolving threats more efficiently. This approach not only enhances the accuracy of intrusion detection but also decreases the time and computational resources required for training models in rapidly changing security landscapes. TL has emerged as a powerful technique in the field of IDS, providing valuable insights and advancements in combating emerging cyber threats. Existing works have employed TL to detect specific novel attack families or focus on specific IoT applications, like the IoH ("Internet of Home"). This article proposes a new and effective framework for detecting various forms of known and new attacks on the basis of TL within IoT networks, surpassing the limitations of existing approaches.The aim of this study is to create and deploy a robust IDS by utilizing TL, knowledge sharing, and model refinement techniques. The researchers assess the accuracy and detection rate of both known and new cyberattack types within IoT networks that possess limited and unbalanced datasets. To achieve this, the authors conducted experiments aimed at unifying the feature space to mitigate disparities when identifying unlabeled Generic samples, which represent zero-day attacks. They accomplished this by utilizing labeled Denial of Service (DoS) samples from the UNSW-NB15 dataset [10].

The authors organize the rest of the article as follows: In Section 2, they provide a comprehensive background and introduce the fundamental concepts of TL. In Section 3, they discuss existing works and research efforts related to intrusion detection and Transfer Learning. Section 4 presents a brief overview of the various machine learnig classifers that are being used in our work .Section 5 presents the novel TL methodology proposed in this study for effective intrusion detection in IoT networks. Finally, they discuss the paper's conclusion and suggested study areas in Section 6.

II. TRANSFER LEARNING CONCEPTS

The limited availability of labeled data in specific fields can be mainly attributed to the high costs associated with data collection, the challenges of manual labeling, and the cold start problem. Conventional ML algorithms face significant difficulties in constructing accurate classifier models in these domains due to the insufficient number of labeled instances. However, transfer learning techniques provide a solution in such situations by enabling the learning of a classifier in a particular domain, even with minimal or no labeled examples. This is achieved by utilizing a sizable database of labeled samples from a relevant source domain, enabling the knowledge and patterns extracted from a source domain to be efficiently moved and used in the target domain.

The process of applying knowledge and abilities from a well-established source domain to a new task in a related target domain is known as transfer learning. It is generally described as follows: Given a "source domain" (D_S) and a related learning task T_S, and a target domain (D_T) and its learning task T_T, the objective of TL is to enhance the "target predictive function" fT(.)learning in D_T by leveraging the knowledge present in D_S & T_S. It's critical to remember that "$D_S \neq D_T$ or $T_S \neq T_T$. The source and target domains are expressed Using a standard formalism :D = {χ, P(X)} for marginal probability distribution P(X) and feature space χ, and T = {Y, f(•)} for the label space Y of a domain [11].

TL could be divided into three dichotomies, each based on different factors. The first dichotomy is determined by the availability of labeled data, while the 2nd dichotomy is based on variations in feature spaces. There are 3 types of TL within the category of labeled data availability: Unsupervised, Transductive and

Inductive TL [11].

Inductive TL applies in situations where there is a limited amount of labeled data in the target domain, regardless of the availability of labeled data in the source domain. The focus is on" leveraging the labeled data available in the target domain to enhance process of learning.On the other hand, Transductive TL occurs when there is no labeled data within target domain, but labeled data is present in the source domain. In this scenario, the aim is to utilize labeled data from the source domain to make estimations specifically for the unlabeled instances in the target domain.Unsupervised TL is applicable when there is no labeled data available in either the source or target domain. The objective here is to extract and transfer knowledge or patterns from the source domain to facilitate unsupervised learning within target domain.These three types of transfer learning address different scenarios on the basis of labeled data availability, allowing for effective knowledge transfer and utilization between domains.

In addition to the dichotomy based on labeled data availability, TL could also be divided into 2 types based on the feature space: homogeneous and heterogeneous TL [12]. Homogeneous TL applies when target and source domains share the same feature space. This means that the input data representations, or features, used in both domains are identical or very similar. In this case, the transfer of knowledge and models between the domains becomes more straightforward, as the underlying representations of the data are consistent. However, heterogeneous TL comes into play when the source feature spaces and target domains differ. This means that the input data representations used in the two domains are distinct or incompatible. In such scenarios, additional techniques, such as feature mapping or transformation, are typically employed to bridge the gap between the feature spaces and enable the effective transfer

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

of knowledge from "source" to "target" domain. By considering both the availability of labeled data and the nature of the feature spaces, transfer learning approaches can be tailored to address specific scenarios and maximize the utilization of knowledge between domains.

III. RELATED WORK

ML was extensively utilized for network intrusion detection, including in IoT environments. However, the datasets currently available in IoT settings are often insufficient to train systems capable of effectively identifying unknown intrusions. As a result, the performance of ML-based intrusion detection systems in IoT networks is hindered, particularly in terms of their ability to detect previously unseen attacks.

Zhao et al. [13] a new transfer learning (TL) method called HeTL is proposed, building upon the HeMap technique introduced in a previous study [10]. The authors tackle the problem by formulating it as a binary classification task. The initial step involves transforming source as well as target data into a shared latent space with spectral transformation. This transformation aims to preserve the original data structure while maximizing the similarity between 2 domains. Subsequently, classification is conducted on the latent features obtained from this transformation. To evaluate the proposed method, the authors derived 3 datasets from the NSL-KDD dataset. In their research, one of these datasets is selected as the source dataset, while one of the remaining 2 datasets serves as the target dataset. This setup allows them to assess the efficiency of the HeTL method in transferring knowledge from "source to target" domain and achieving improved classification performance within target domain.

Sameera et al. [14] applied TL to intrusion detection systems (IDS) to detect zero-day attacks and minimize the false positive rate (FPR). They specifically focused on detecting R2L ("Remote-To-Local") attacks and designed a system to detect unlabeled R2L attacks within the dataset of NSL-KDD. To accomplish this, they leveraged labeled DoS (Denial of Service) attacks available in the dataset, and attained an impressive accuracy of 89.79 percent and a low FPR of 0.15% by applying transfer learning techniques. This approach represents an improvement of 11.79% compared to previous feature-based transfer learning methods.

Singla et al. [15] suggest a system that focuses on identifying particular families of new attacks by utilizing transfer learning (TL) techniques. The aim is to "transfer knowledge from source to target domain, even when the target model has limited training data available. In their implementation, Singla et al. employ a TL model consisting of two deep neural networks (DNN). One DNN comprises two regular densely connected layers, while the other DNN consists of five such layers. The researchers divide the UNSW-NB15 dataset [11] into 2 parts: (i) a source dataset comprising various types of attacks, and (ii) a target dataset consisting of a novel type of attack. By comparing the TL solution" with a baseline deep learning (DL) model trained from scratch, Singla et al. evaluate the accuracy improvement achieved by the TL approach. The findings indicate that TL solution enhances the accuracy by a range of 3.2% to 19.1%, based on the type of novel attack being detected.

Vercruyssen et al. [16] make two key assumptions: that anomalies are rare occurrences and that they exhibit unexpected behavior. They propose a transfer learning approach based on instance-based reweighting, which aims to match examples from above-mentioned domain for "time series anomaly detection". The issue is formulated as a binary classification task. To make decisions regarding the transfer of instances, the authors introduce 2 decision functions: the "density-based transfer decision function" and the "cluster-based transfer decision function." These functions enable the system to determine whether an instance should be moved from source to target domain based on its density or its clustering characteristics.

Zahra Taghiyarrenani et al. [17] present a novel TL method that focuses on mapping the source as well as target datasets into a shared feature space. They achieve this by employing a manifold alignment approach that involves four matrices: target structural, source structural, dissimilarity, and similarity matrix. By aligning the manifolds of the source & target datasets, the suggested TL method aims to enhance the transferability of knowledge between the domains. Once the datasets are mapped into the common feature space, the problem is expressed as a binary classification task. Support Vector Machines (SVMs) are then utilized to identify the target examples into either attack or normal classes. To evaluate the effectiveness of their TL approach, the researchers conduct experiments using the KDD 99 and Kyoto2006+ datasets. These datasets serve as the basis for assessing the suggested approach performance e in terms of classification accuracy and the ability to detect attacks.

Ahmadi et al. [18] focus on detecting DoS attacks in a cloud environment by leveraging attack knowledge from a "non-cloud environment". Their approach involves utilizing common features from the source (non-cloud) and target (cloud) domains to enhance the detection of DoS attacks. To transfer knowledge between the two domains, the authors employ a TL method called Relation-based TL. This approach relies on manually identifying and defining pre-defined relations between source as well as target domains. By leveraging these relations, the knowledge from the non-cloud environment is transferred to the cloud environment to enhance detection capabilities.

Fan et al. [19] explain the combination of TL and federated learning in the context of 5G Internet of Things (IoT) environments. They suggest a federated framework that enables secure data aggregation from various IoT networks. The goal is to develop personalized IDS for every IoT network using TL. To implement TL, the researchers utilize Convolutional Neural Networks (CNNs). They leverage the knowledge gained from a base dataset, specifically the CICIDS2017 dataset [20], and transfer this knowledge to different custom target datasets representing diverse IoT networks. By adapting the intrusion detection

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

models to the specific characteristics of each IoT network, the researchers aim to improve the accuracy and effectiveness of intrusion detection.

Mehedi et al. [21] introduce a novel approach for intrusion detection in heterogeneous IoT networks using a residual neural network based on Deep Transfer Learning (DTL). They create a custom dataset by aggregating data from seven different IoT sensors, enabling the identification of nine distinct forms of attacks including ransomware, XSS, scanning, PCA, backdoor, MITM, data injection, DDoS, and DoS. To effectively detect intrusions, the researchers employ a CNN-based model with residual connections. The model is trained using the collected dataset, and its performance is evaluated based on its ability to accurately classify different attack types. The overall accuracy achieved by the CNN-based model is reported to be 87%.

Guan et al. [22] capitalize on previous advancements in traffic classification [23] to develop a deep TL approach for network classification within IoT environments. This approach is designed to address the challenge of limited labeled data and devices with constrained computing capabilities. The researchers leverage the power of EfficientNet [24] and BiT ("Big Transfer") [25], two well-established models that have shown exceptional performance in image recognition tasks using transfer learning. To evaluate their solution, Guan et al. employ the 10% USTC-TFC2016 labeled dataset [26], which represents a scenario with scarce labeled data. The proposed method achieves impressive accuracy rates, with 96.22% for BiT and 96.40% for EfficientNet. These results indicate the effectiveness of the deep transfer learning approach in overcoming data scarcity and limited computing resources for accurate network classification in IoT environments.

IV. MACHINE LEARNING APPROACHES

In this section, the supervised ML methods that were applied in this work are brifly overviewd.

A. Dession Tree (DT):

A DT [27] is a method that aims to divide individuals into groups based on their similarities in relation to the target variable. The method constructs a tree structure that represents hierarchical links between variables. The decision tree-building process is iterative. At every iteration, the algorithm selects an explanatory variable that provides the best separation of individuals into distinct groups. This variable is used as a splitting criterion to create branches in the tree, resulting in subsets of individuals that exhibit similar characteristics. The process continues recursively, with each subset of individuals being further divided based on the most informative variables until a stopping criterion is met. The stopping criterion is reached when no further splits can be made, indicating that the tree has captured the patterns and relationships in the data to the best extent possible.

B. Random Forest (RF):

RF algorithms are powerful techniques used for classification and regression tasks [28,29]. They consist of an ensemble of multiple Decision Trees that serve as individual predictors. The fundamental concept behind Random Forest is to generate a significant number of Decision Tree models instead of relying on a single optimized model. Each DT is trained on a random subset of data, using a random selection of features. This introduces diversity in the trees' predictions. During the prediction phase, each Decision Tree in the forest independently provides its prediction. The class that attains the most votes from the individual trees is chosen as the final forecast in classification problems investigated by the majority vote. In the case of regression tasks, the estimated value is generally calculated as the average or median of the estimated values from every tree. By combining the predictions from multiple trees and considering their collective wisdom, Random Forests can provide robust and accurate predictions. They are known for their ability to handle complex datasets, mitigate overfitting, and provide insights into variable importance.

C. Extra-Trees

ET also-referred to as "Extremely Randomized Trees", is another supervised machine learning method that utilizes multiple Decision Trees (DTs) for decision-making. ET can be applied to both classification & regression problems. Unlike RF, where every DT is built from a subset of the training set, ET fits individual DTs using the entire training set. Additionally, the ET approach randomly selects split points for each node, further enhancing the randomness of the tree construction process [30, 21].

D. Multi-Level Perceptron (MLP)

A MLP [33] is a kind of ANN ("Artificial Neural Network") made up of many linked layers of perceptron-like neurons. As a feedforward neural network, it only allows only one direction of data flow—from the I/P layer to the O/P layer. An I/P layer, one or more hidden layers, and an O/P layer make up the MLP. The neurons in neighboring layers are completely linked, which indicates that each neuron in one layer is linked to every neuron in the layer above it. Each layer is made up of a collection of neurons, also known as nodes. Each neuron in an MLP gets inputs from the neurons in the layer below, processes the weighted sum of these inputs using an "activation function", and then creates an output. The network may learn complex patterns as well as correlations in the data due to the activation function's introduction of non-linearity. The MLP is trained using a process called backpropagation, which involves forward propagation of input data through the network to produce an output, comparison of the output with the desired output, calculation of the error, and adjustment of the weights in the network to minimize the error. This process is repeated iteratively until the network learns to make accurate predictions. MLPs are widely utilized for various tasks like classification, regression, and pattern recognition.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

They have proven to be effective in solving complex problems and can handle large amounts of data. Nevertheless, they may suffer from overfitting if the model becomes too complex or if the dataset is insufficient.

E. Principal Component Analysis (PCA):

PCA is a dimensionality reduction technique used in data analysis and machine learning. Here are the key mathematical formulations and formal definitions for PCA:

Given: A dataset represented as a matrix X, where each row corresponds to a data point, and each column corresponds to a feature.

Objective: Find a set of orthogonal vectors (principal components) in the original feature space such that when data points are projected onto these components, the maximum variance is retained in the first principal component, the second maximum in the second component, and so on.

Mean-Centering:

Compute the mean (centroid) of the data:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x^i \text{----------(1)}$$

Subtract the mean from each data point to center the data:

$$x_i' = x_i \ -\mu \text{ ----------(2)}$$

Covariance Matrix:

Compute the covariance matrix of the mean-centered data:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i'.\ x_i'^{T} \text{---------- (3)}$$

Eigenvalue Decomposition:

Solve the eigenvalue problem for the covariance matrix:

$$\sum V_{J=} \lambda_j V_j \text{----------(4)}$$

Where:

- λj is thej-th eigenvalue.
- Vj is the corresponding j-th eigenvector.

Sorting Eigenvalues and Eigenvectors:

Sort the eigenvalues in descending order:

$$\lambda_{1\geq}\lambda_{2\geq} \dots\dots\dots\dots \lambda_p \text{----------(5)}$$

Correspondingly, sort the eigenvectors accordingly:

$$V_{1,}V_2 \dots\dots\dots\dots V_p \text{ ----------(6)}$$

Selecting Principal Components:

Choose the top k eigenvectors to retain k principal components. Typically, k is determined based on how much variance you want to retain (e.g., 95% of the total variance).

Projection:

Project the mean-centered data onto the selected principal components to obtain the reduced data matrix:

$$Y=X'V_k \text{----------(7)}$$

Where:

- Y is the reduced data matrix.
- X' is the mean-centered data.
- V_k is a matrix consisting of the first k eigenvectors as columns.

Alioritham1 : PCA Algorithm

```
function PCA(X, num_components):
    # X: Input data matrix, where each row represents a data point, and each column represents
        a feature
    # num_components: Number of principal components to retain
    # Step 1: Mean centering
        mean = compute_mean(X)
        X_centered = center_data(X, mean)
    # Step 2: Compute the covariance matrix
        covariance_matrix = compute_covariance_matrix(X_centered)
    # Step 3: Compute the eigenvalues and eigenvectors of the covariance matrix
        eigenvalues, eigenvectors =
                compute_eigenvalues_and_eigenvectors(covariance_matrix)
    # Step 4: Sort eigenvalues and eigenvectors in descending order
        eigenvalues, eigenvectors = sort_eigenvalues_and_eigenvectors(eigenvalues,
                eigenvectors)
    # Step 5: Select the top 'num_components' eigenvectors
        selected_eigenvectors = eigenvectors[:num_components]
    # Step 6: Project the data onto the selected eigenvectors
        reduced_data = X_centered.dot(selected_eigenvectors.T)
        return reduced_data
```

PCA provides a way to reduce the dimensionality of your data while preserving the most important information, often used for data visualization, noise reduction, or feature selection in various data analysis and machine learning tasks. Here's a step-by-step overview of the PCA Algorithm1.

V. TL-BASED PROTOTYPE FOR INTRUSION DETECTION

The proposed methodology aims to address the challenge of detecting zero-day attacks, which are often unlabeled or scarcely labeled. To achieve this, transfer learning (TL) is employed, where classifiers are built in the unlabeled target domain using a related source domain that has a significant number of labeled examples. This categorizes the suggested methodology as transductive TL. Zero-day attacks are designed to evade detection based on known attack signatures, and they possess a different set of relevant features compared to known assaults. Consequently, the suggested methodology also falls into the class of heterogeneous TL, as it deals with the differences in feature space between source as well as target domains. The labeled instances of known attacks make up the relevant source domain in the TL issue, whereas the zero-day attack data relates to the target domain. Since there are differences in the feature space between these domains, heterogeneous transductive TL is used. A binary classification challenge is

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

how the suggested TL-based prototype for "zero-day attack detection(ZDAD)" approaches the issue.

The methodology, as illustrated in Figure 1, involves combining the source as well as target domains with their heterogeneous feature spaces. To bring them into the same feature space, an orthogonal transformation is applied using Principle Component Analysis (PCA) with eigenvalues and eigenvectors. This transformation ensures that both domains meet the necessary condition for applying a classifier.

From the combined transformed data, source training data and target testing data are extracted. These extracted sets are then used as input for the classification process, which outputs labels (attack/normal) for the target data examples.

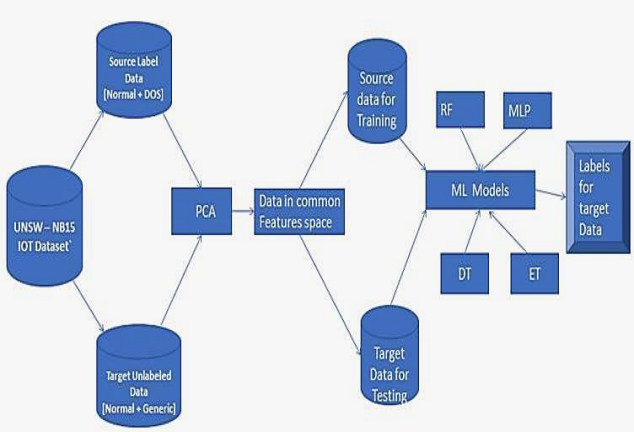

Fig 1: The proposed ZDAD -TL technique's block diagram

A. Data set preparation:

The suggested approach utilizes the UNSW-NB15 dataset, which is a widely recognized benchmark dataset used by several investigators for implementing IDS in IoT. Instances from nine different assault groups are included in this collection: Normal, Generic, DoS, Fuzzers, Reconnaissance, Backdoors, Analysis, Shellcode, and Worms. For the purpose of this study, two sub-datasets are derived from the dataset of UNSW-NB15: the DoS sub-dataset and the Generic sub-dataset. The DoS sub-dataset contains the normal cases along with all the DoS attack instances. The R2L sub-dataset, on the other hand, contains all normal instances along with all the Generic attack instances. By using these derived sub-datasets from the UNSW-NB15 dataset, the proposed method can train and evaluate the intrusion detection model specifically for DoS and Generic attacks.

B. Experimentation and Result Analysis:

To conduct experiments and evaluate the model, a small set of labeled data is needed from the test set. A classifier is constructed using a training set that consists of a sufficiently significant number of labeled instances from a source domain. To test the proposed TL method, the authors utilized the Generic dataset as the target dataset and the DoS dataset as the source dataset. Below are the details of their experimentation.

- **D_S = DoS+ Normal**
- **D_T = Generic + Normal**
- **Source label space: {DOS, Normal}**
- **Target label space: {Generic, Normal}**

The step by step procedure to develop the ***ZDAD*** algorithm2 model is presented below:

In the proposed TL approach, the Generic dataset is considered the target dataset, while the DoS dataset is utilized as the source dataset. The target label space consists of two classes: Generic and Normal, while the source label space includes DoS and Normal.

Since the objective is to identify zero-day attacks, the generic dataset class labels are removed and kept aside for performance evaluation of the TL approach. As a result, the target dataset becomes unlabeled, while the source dataset remains labeled.

After preparing the dataset, the DoS and Generic datasets undergo Principal Component Analysis (PCA), resulting in a transformation into a 12-dimensional feature space. The transformed data is then split into a training set, comprising the DoS dataset, and a testing set, comprising the Generic dataset. The training and testing instances are subjected to classification using various classifiers: Random Forest, Decision Tree, Multi-level Perceptron, and Extra-Tree.

The classification process yields impressive results, Random forest with an accuracy of 99.24% and a FPR of 0.02% in effectively detecting zero-day attacks (Generic). These outcomes are summarized in Table 1. It is worth noting that no comparisons are made with No_TL methods since the assumption is that the target data (Generic) is entirely unlabeled, which differs from the assumptions of No_TL approaches.

Alorithm 2 : ZDAD -TL Algorithm

Step 1: source domains

D_S = DoS + Normal

Step 2 : Target domains

D_T = Generic + Normal

Step 3: Combine the source and target domains

combined_data = combine(source_data, target_data)

Step 4: Perform Principle Component Analysis (PCA) for orthogonal transformation

transformed_data=apply_PCA(combined_data)

Step 5: Split the transformed data into source train and target test sets

source_train_data, target_test_data =split_data(transformed_data)

Step 6: Train the classifier using source train data

classifier = train_classifier(source_train_data)

Step 7: Classify the target test data using the trained classifier

target_labels = classify_data(classifier, target_test_data)

Step 8: Output the labels for target data instances

output(target_labels)

Table 1: Zero-day attack detection Accuracy

Classifier	Accuracy	Dataset used	Precision	Recall	F1-Score	TPR	FPR
DT	93.07	DOS→Generic	0.83	0.99	0.90	0.99	0.16
MLP	95.52	DOS→Generic	0.88	0.99	0.93	0.99	0.11
RF	99.24	DOS→Generic	0.97	0.99	0.98	0.99	0.02
ET	98.37	DOS→Generic	0.95	0.99	0.97	0.99	0.04

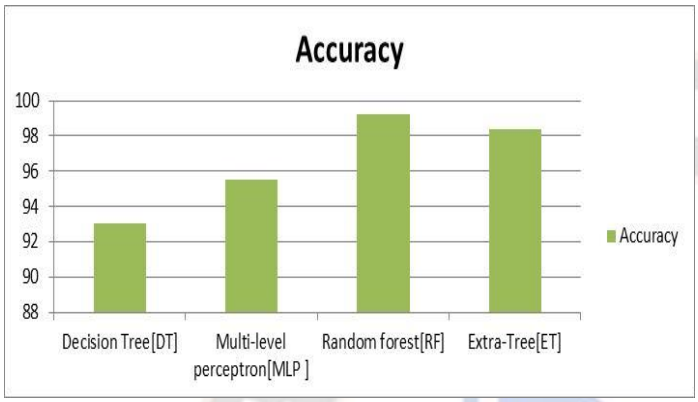

Fig 2: Accuracy of selected classifiers when using the dataset of UNSW-NB15

VI. CONCLUSION

As the digital age advances, cyber threats are increasing, and this has led to a growing interest in IDS. Presently, researchers are actively exploring how TL can effectively identify zero-day attacks and reduce False Positive Rates (FPRs) in IDS. TL provides methodologies for constructing classifiers within a target domain, even when limited or no labeled data is available, by leveraging knowledge extracted from the source domain(s). Researchers are actively exploring how TL can be used in IDS to detect zero-day attacks and reduce FPR. TL provides methodologies for constructing classifiers in target domains that have limited or no labeled data. It achieves this by leveraging knowledge obtained from related source domains. The authors of this research article used TL ideas to identify unlabeled generic (zero-day) attacks using a dataset of "UNSW-NB15". The classifiers tested, Random Forest (RF) achieved the highest accuracy of 99.24%. This indicates that the RF model performed exceptionally well, accurately classifying 99.24% of instances in the dataset. The high accuracy reflects the model's ability to effectively distinguish between normal and attack instances, making it a reliable and robust option for IDS applications. Furthermore, RF demonstrated an impressively low FPR of 0.02%. A low FPR is crucial in IDS as it indicates a reduced rate of false positives, where normal instances are incorrectly identified as attacks. This minimizes unnecessary alarms and helps ensure that security analysts focus on genuine threats, enhancing the efficiency and effectiveness of the IDS. The Multi-level perceptron (MLP) and Extra-Tree (ET) classifiers also showcased strong performances with accuracies of 95.52% and 98.37%, respectively. Although slightly lower than RF, these classifiers still exhibited high accuracy rates, indicating their competence in detecting zero-day attacks. However, it's worth noting that both MLP and ET had slightly higher FPRs compared to RF. MLP recorded an FPR of 0.11%, and ET had an FPR of 0.04%. While these rates are reasonable, they suggest a marginally higher tendency to generate false positives compared to RF. The Decision Tree (DT) classifier achieved an accuracy of 93.07%, making it the lowest accuracy among the tested classifiers. Additionally, it had an FPR of 0.16, the highest among the classifiers, indicating a higher rate of false positives.

REFERENCES

[1] Internet Security Report. Available online: https://www.watchguard.com/wgrd-resource-center/security-report-q3-2020 (accessed on 27 May 2022).
[2] Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K. Consumer IoT: Security Vulnerability Case Studies and Solutions. IEEE Consum. Electron. Mag. 2020, 2, 17–25. [CrossRef]
[3] Kilincer, I.; Ertam, F.; Sengur, A. Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Comput. Netw. 2021, 188, 107840. [CrossRef]
[4] Fadlullah, Z.M.; Tang, F.; Mao, B.; Kato, N.; Akashi, O.; Inoue, T.; Mizutani, K. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun. Surv. Tutor. 2017, 19, 2432–-2455. [CrossRef]
[5] Thamilarasu, G.; Chawla, S. Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things. Sensors 2019, 19, 1977. [CrossRef]
[6] Rodríguez, E.; Otero, B.; Gutiérrez, N.; Canal, R. A Survey of Deep Learning Techniques for Cybersecurity in Mobile Networks. IEEE Commun. Surv. Tutor. 2021, 23, 1920–1955. [CrossRef]
[7] Wu, P.; Guo, H.; Buckland, R. A Transfer Learning Approach for Network Intrusion Detection. In Proceedings of the IEEE 4th International Conference on Big Data Analytics ICBDA, Suzhou, China, 15 March 2019; pp. 281–285.
[8] Ruder, S.; Peters, M.; Swayamdipta, S.; Wolf, T. A Transfer Learning in Natural Language Processing Tutorial. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, Minnesota, 1 June 2019; pp. 15–19.
[9] Kasthurirangan, G.; Khaitan, S.; Choudhary, A.; Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. Constr. Build Mater. 2017, 157, 322–330
[10] Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the Military Communications and Information Systems Conference MilCIS, Canberra, Australia, 10 November 2015; pp. 1–6
[11] Pan, SinnoJialin, and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, no. 10 2010, 1345-1359
[12] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang, "A survey of transfer learning", Journal of Big Data 3.1 (2016): 9.
[13] Zhao, Juan, Sachin Shetty, and Jan Wei Pan, "Feature-based transfer learning for network security", In MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM), pp. 17-22. IEEE, 2017
[14] Sameera, N.; Shashi, M. Transfer Learning Based Prototype for Zero-Day Attack Detection. Int. J. Eng. Adv. Technol. 2019, 8, 1326–1329.
[15] Singla, A.; Bertino, E.; Verma, D. Overcoming the Lack of Labeled Data: Training Intrusion Detection Models Using Transfer Learning. In Proceedings of the IEEE International

Conference on Smart Computing SMARTCOMP, Washington, DC, USA, 12 June 2019; pp. 69–74
[16] Vercruyssen, Vincent, WannesMeert, and Jesse Davis, "Transfer Learning for Time Series Anomaly Detection", In IAL@ PKDD/ECML, pp. 27-36. 2017.
[17] Zahra Taghiyarrenani, et,al. "Transfer Learning based Intrusion Detection", International Conference on Computer and Knowledge Engineering (ICCKE 2018), October, pp. 25-26, 2018.
[18] Ahmadi, Roja, Robert D. Macredie, and Allan Tucker, "Intrusion Detection Using Transfer Learning in Machine Learning Classifiers Between Non-cloud and Cloud Datasets", In International Conference on Intelligent Data Engineering and Automated Learning, pp. 556-566. Springer, Cham, 2018
[19] Fan, Y.; Li, Y.; Zhan, M.; Cui, H.; Zhang, Y. IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT. In Proceedings of the IEEE 14th International Conference on Big Data Science and Engineering BigDataSE, Guangzhou, China, 1 January 2021; pp. 88–95.
[20] Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A.A. New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy ICISSP, Funchal, Portugal, 24 January 2018; pp. 108–116
[21] Mehedi, S.T.; Anwar, A.; Rahman, Z.; Ahmed, K.; Islam, R. Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-based Approach. IEEE Trans. Ind. Inf. 2022, 1, 1–12.
[22] Guan, J.; Cai, J.; Bai, H. Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems. Int. J. Mach. Learn. Cyber 2021, 12, 3351–3365.
[23] Sun, G.; Liang, L.; Chen, T.; Xiao, F.; Lang, F. Network traffic classification based on transfer learning. Comput. Electr. Eng. 2018, 69, 920–927.
[24] Tan, M.; Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning PMLR, Long Beach, CA, USA, 15 June 2019; pp. 6105–6114.
[25] Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; Houlsby, N. Big Transfer (BiT): General visual representation learning. In Proceedings of the European Conference on Computer Vision ECCV, Glasgow, UK, 28 August 2020; pp. 491–507.
[26] USTC-TFC2016. Available online: https://github.com/yungshenglu/USTC-TFC2016/ (accessed on 27 May 2022)
[27] Mr Brijain, R Patel, Mr Kushik, and K Rana. A survey on decision tree algorithm for classifcation. International Journal of Engineering Development and Research, IJEDR, 2(1), 2014.
[28] Breiman L. Random forests Machine learning. 2001;45(1):5–32.
[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Random forests. In The elements of statistical learning, pages 587–604. Springer, 2009.
[30] Ahmad, M.W.; Reynolds, J. and Rezgui; Y. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. Journal of cleaner production, 2018, 203, 810–821.
[31] Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O. and Alazzawi, A.K. AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access, 2020, 8, 142532–142542.