From One to Many: Transfer Learning's Expansive Reach in Network Attack Detection

John Gallagher Andrew Song
University of Virginia
jjg5fg, ajs4aw@virginia.edu

Abstract

To enhance threat detection, this project investigates the application of transfer learning in network security. Two models were developed as part of the project to examine how threat detection is impacted by pre-trained models. Specifically, the project evaluated how little data may be required to produce an effective model that can be applied to new threats. A dataset of traditional network traffic with numerous attacks was used to train one model, whereas IoT traffic with a single kind of attack (DoS) was used to train the other model. To test the impact of these two models' capacity to be "transferred" to forecast attacks in previously uncovered data, they were applied to additional datasets. The model trained on more attacks obtained significantly higher accuracy, indicating that a substantial amount of training data is required for successful transfer learning.

1 Introduction

The prevalence of network attacks today continues to rise as the number of connected devices expands. In 2023, a record 12.5 billion dollars was lost to internet crime, with business email compromise crimes totaling 2.9 billion [9] This is only getting more and more complex, with minimally skilled criminals able to get the skills needed for attacks through various new technologies like LLMs [11].

Although network security teams are getting better and better at detecting and mitigating known threats, attack patterns change rapidly, especially as zero-day attacks become more and more prevalent, with a big increase in 2023 compared to 2022. [10]This evolution presents challenges in data collection and labeling, which is time-consuming and resource-intensive for network security and data science teams. Transfer learning, a machine learning technique that leverages pre-trained models to adapt to new and unseen data, can address these challenges by reducing both the time and cost associated with developing robust models.

This adaptability is crucial for identifying emerging threats and enhancing network security. This paper examines the application of transfer learning to network security, focusing on building several models to test a variety of attacks across different systems. The aim is to provide a comprehensive evaluation of the effectiveness of transfer learning in detecting novel network attacks, as well as establish how much or little data is required to build an effective model

2 Literature Review

Many works have explored and implemented the idea of applying transfer learning to detect network attacks. In 2019, "Transfer learning for detecting unknown network attacks" by Zhao et al. was published, implementing an innovative "transductive transfer learning" approach that aimed to mimic the human's transitive inference ability to extend what has been learned in one domain to another domain. [14] Other works have applied transfer learning to specific attacks. For example, "Deep transfer learning for IoT attack detection" by Vu et al. aimed

to use transfer learning to specifically detect IoT attacks. [13]

Ultimately, with our work, while we aim to do similar things as the works above and many other related works, we specifically aim to achieve two things with our model. Firstly, we aim to detect unknown network attacks. We want to be able to use transfer learning to act quickly against threats like zero-day attacks, which require the ability to detect unknown threats. Secondly, we want to be able to create an effective transfer learning model given a limited amount of training data. Realistically, one cannot always expect to have a large amount of data to train on, thus, to better mimic real-world scenarios, we aim to create a model that does not rely on a huge training set.

3 Data Description

The project looks at three different datasets for the training, validation, and testing of the models built with transfer learning. Specifically, we look at two datasets from attacks on the Internet of Things (IoT): ACI-IoT-2023 and BoTNeTIoT-L01, and one dataset of traditional network attacks, CICIDS2017. All the datasets were available to the public and accessed from Kaggle.

3.1 ACI-IoT-2023

The Army Cyber Institute (ACI) Internet of Things (IoT) Network Traffic Dataset 2023 (ACI-IoT-2023) is a dataset that focuses on network attacks related to IoT devices. The dataset is a week's worth of data, with days either looking at a specific type of attack or being benign. The attack types include Reconnaissance (Recon), Denial of Service (Dos), Brute Force, and Spoofing. The novelty in this dataset is the dynamic mix of IoT devices and how they connect to a traditional home network. Kaggle had the PCAP files all converted to one massive CSV which allowed for ease of use in Python. [7] [3] Figure 1 shows the range of devices captured in the dataset.

Figure 1: Devices from ACI dataset

3.2 BoTNeTIoT-L01

BoTNeTIoT-L01 is a dataset from Kaggle that reduces the redundancy of the original dataset detection of IoT botnet attacks, BoTNeTIoT, by choosing features of 10-second time windows only. This dataset is based on nine IoT devices using WireShark in a local network switch. It contains 23 features and two Botnet attacks: Mirai and Gafgyt, a type of Bashlite attack. These are attacks unique to IOT devices and bring in a unique attack for evaluation of transfer learning. [2]

3.2.1 Botnet attacks

Bashlite and Mirai are malware that infects smart devices that run on a stripped-down Linux processor, or ARC, turning them into a network of remotely controlled bots or "zombies". This network of bots, called a botnet, is often used to launch DDoS attacks. Mirari is built off of Bashlite as its predecessor. [5] Figure 2 from the paper by Marzano et al is a diagram demonstrating how a botnet attack works. [6]

3.3 CICIDS2017

CICIDS2017 is a dataset containing common attacks. It looks at the results of network traffic analysis using CICFlowMeter, containing labeled flows based on the time stamp, source, and destination IPs, source and

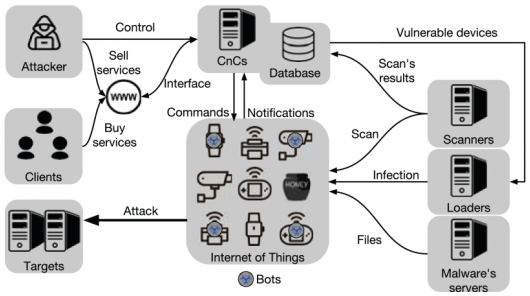

Figure 2: Botnet Attack Diagram

destination ports, protocols, and attack. This information was available in CSV files. The data is based on a 5-day window with benign traffic mixed with the following attacks: Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS. [4] The testbed architecture for the data is shown in Figure 3 [8]

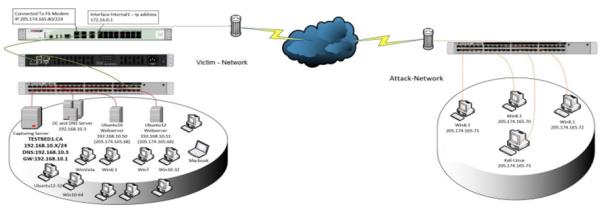

Figure 3: CICIDS2017 testbed architecture

4 Methodology

One of the key aspects of a transfer learning model is to ensure its flexibility when applied to different datasets. Two different models were created: one model using ACI-IoT as the main training dataset on just DoS attacks and "transferred" to the other two datasets and then another model using CICIDS2017 with all the attacks as its main dataset and "transferred" to the ACI-IoT and BoTnetIoT-L2 datasets.

4.1 Model Building

To begin the construction of our model for transfer learning, we built a neural network from scratch to effectively classify new and unseen data by recognizing complex patterns learned during training. The model characteristics were the same for both neural networks built. The neural network was built using Keras package in Python, and the input layer consisted of 256 neurons with a 'relu' function, a common configuration for neural networks. Normalization and dropout were applied which stabilizes the model and prevents overreliance on specific data parts by normalizing input values and randomly ignoring some neurons during training. For the hidden layer, the neurons drop in half each time with a dropout of .50, so that half of the nodes are randomly dropped in each weight update cycle, helping to prevent overfitting. The output layer uses softmax to output probabilities for each class, making the final categorization based on the highest probability. For the optimizer, Adam was used with a learning rate of .001, and the loss function used was sparse categorical cross entropy which evaluates how well the model's predictions match the actual categories. These parameters result in a model that combines batch normalization and dropout to learn robustly and accurately, ensuring reliability even with varied data. Figure 4 shows the model summary. [1]

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 256)               20224

 batch_normalization (Batch  (None, 256)               1024
 Normalization)

 dense_1 (Dense)             (None, 128)               32896

 dropout (Dropout)           (None, 128)               0

 batch_normalization_1 (Bat  (None, 128)               512
 chNormalization)

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 15)                975

=================================================================
Total params: 63887 (249.56 KB)
Trainable params: 63119 (246.56 KB)
Non-trainable params: 768 (3.00 KB)
```

Figure 4: Keras Model Arcitecture

4.1.1 ACI-IoT

The main purpose of the model trained on the ACI-IoT data was to make it as flexible as possible to establish a baseline for how many attacks an effective model needed to be trained on. The ACI-IoT was subsetted to just be trained on DoS attacks and benign traffic. The data was cleaned in a traditional machine learning way: unnecessary variables were dropped, and categorical variables like network connection were hot encoded. The data was normalized for the model and split into 60/40 train test split, to ensure a lack of overfitting. Training the model on 10 epochs with a batch size of 32 and a validation split of .2, the model was able to accurately label the attacks in ACI-IoT with .997 accuracy. The model was then saved to disk to be loaded for transfer learning.

4.1.2 CICIDS2017

As opposed to the model in ACI-IoT which only trained on DoS attacks, the model trained on CICIDS was trained using all of the different types of attacks in the dataset. This allowed for comparison to the baseline of one attack on IoT devices as opposed to numerous attacks in a traditional sense. For CICIDS, the data was split into eight csvs, each corresponding to the morning/afternoon of the various days. Data was read in and merged into one giant data frame to train the model on. As with ACI, traditional ML preprocessing was done, with data scaled, categorical one hot encoded, and split into a 60/40 split. Applying the same Keras model above, with a batch size of 32, and 10 epochs, the model was able to obtain a .984 test accuracy. This model was also saved to disk to be loaded for transfer learning.

4.2 Transfer Learning Methodology

Transfer learning is the process of taking a pretrained model and applying it to a new dataset. To start, the model was read in from disk. It was then modified by discarding the original output layer, which was specific to the original task. A new output layer was created with a new number of predictive classes using the same softmax function. The input layer was altered to change the amount of features the model is retrained on. The other layers were frozen, which ensured that the model was only trained on the bottom layer and that the weights from the original model were used. The model was then recompiled using Adam as the optimizer and a learning rate of.0001 to make cautious updates. New data was processed using the same techniques above and the data was split on a more traditional .80/20 train test split. The model was fit using 50 epochs and a batch size of 500, ensuring more training on data not native to the original model. This was run for the other two datasets not used to train the original model. So, for the ACI model, CICIDS2017 and BoTnetIoT were applied for transfer learning. For CICIDS2017, ACI and BoTNeTIoT were applied for transfer learning. [12]

5 Results

The following sections describe the results of taking our two models and transferring them to the other datasets.

5.1 ACI Model

The ACI model, trained exclusively on Denial of Service (DoS) attacks and benign traffic from the ACI-IoT dataset, was tested on two datasets for transfer learning on CIDS2017 and BotNetIoT.

5.1.1 Transfer to CICIDS2017

The ACI model achieved an accuracy of 83.2 on the CICIDS2017 dataset. This performance is better than expected given that the model was initially trained on a much narrower set of attacks. The CICIDS2017 dataset encompasses a broader range of attacks, including Brute Force, DDoS, and Web attacks, which the ACI model had not encountered in its training. This result indicates the potential of transfer learning in expanding a model's capacity to recognize previously unseen threats.

5.1.2 BotNetIoT

When applied to the BotNetIoT dataset, the ACI model's accuracy fell to 71.5. The primary challenge appeared to be the limited training data, which did not cover the unique characteristics of BotNetIoT's Mirai and Gafgyt attacks. This suggests that while transfer learning can extend a model's applicability, its effectiveness is closely tied to the diversity of its training data.

5.2 CICDS2017 Model

The CICIDS2017 model, trained on a comprehensive set of attack types across a traditional network environment, was tested on the ACI-IoT and BotNetIoT datasets

5.2.1 ACI-IoT

The CICIDS2017 model performed exceptionally well on the ACI-IoT dataset, with a test accuracy of 96. This result is remarkable because it approaches the accuracy of models trained directly on ACI data. The high accuracy demonstrates the robustness of models trained on diverse datasets, as the patterns learned from various attack types can translate well to new attack environments.

5.2.2 BotNetIoT

For the BotNetIoT dataset, the CICIDS2017 model achieved an accuracy of 83. Despite being trained primarily on traditional network attacks, the model could still effectively identify botnet attacks. This suggests that the training on a wide variety of attacks in the CICIDS2017 dataset provided sufficient exposure to different attack patterns, which improved the model's adaptability to different network contexts.

6 Discussion

This project aimed to compare the effects of transfer learning in enhancing network attacks by comparing two models: one trained on a specialized dataset on a single type of attack (ACI) and another trained on a comprehensive dataset covering a wide variety of traditional attacks. The results highlight key insights into the applicability of transfer learning in network security.

6.1 Model Versatility and Adaptability

The CICIDS model showed superior adaptability achieving extremely high accuracy when applied to ACI dataset, and moderate accuracy on BotNetIoT. The adaptability is attributed to the model being trained on a large spectrum of attacks allowing it to learn more patterns. This adaptability is promising, given the new threats network security experts face day to day.

The ACI model trained on a single attack struggled with datasets containing different attack types, especially botnet attacks. This shows that the more information the model is fed the better it is at generalizing. This gap in performance shows the importance of data diversity in training a model and detecting new and novel threats.

6.2 Applications

This study shows the potential of leveraging transfer learning to enhance the adaptability of network security models. By leveraging pre-trained models, organizations can reduce the time and computation needed to train new models each time. Given the rapid adaptation needed to counter emerging threats, time savings can be huge. However, for large-scale applications, larger and more diverse training would need to be done to ensure the robustness of the model.

6.3 Limitation and Future Work

6.3.1 Limitations

Transfer learning showed promise however it does have some challenges. Obtaining a large, well-labeled dataset to cover the range of attacks a network security expert would need takes a lot of work. Transfer learning also may not work well in unique and different network environments. As a result, transfer

learning could struggle with zero-day attacks that are not variations of known attacks.

6.3.2 Future Works

This project aimed to investigate the initial impact of transfer learning on network attacks and how much data would be needed to create an effective model. More work on the fine-tuning of the model and evaluation should be done. However, this could result in loss of adaptability due to overfitting. More work could look at 5G network traffic and how those attacks would be evaluated. Finally, to get past the issue of data, Generative Adversarial Networks (GAN) can be explored to create network traffic to train a model on, giving the model a broader range of attacks.

7 Conclusion

Transfer learning's potential to improve threat detection in network environments has been brought to light by evaluations conducted in this project. The project presents two models: one tailored for an attack-focused dataset (ACI) and another trained on a broad dataset covering several attacks (CICIDS). The project demonstrates how the model's performance is affected by the variety and volume of training data.

Because of its diverse training data, the CICIDS model was able to capture patterns for generalization with unseen data, demonstrating flexibility and excelling in the detection of unique threats across diverse datasets. This adaptability highlights the promise of transfer learning in network security by leveraging trained models.

On the other hand, the ACI model demonstrated limits as it was created for a particular network and an attack scenario. This discrepancy in performance emphasizes how important different training datasets are for the model in transfer learning situations.

Transfer learning poses difficulties in addition to solutions. One limiting factor is still the lack of readily available, network-specific annotations for data. Network environments may impose restrictions on transfer learning. Subsequent investigations ought to concentrate on enhancing models to achieve an equilibrium between flexibility and precision while tackling novel developments such as 5G networks and growing security risks. GANs, or Generative Adversarial Networks, may provide a way around data limitations. To determine their effectiveness, further evaluation is necessary.

Ultimately, transfer learning shows promise as a means of strengthening network security by reducing the need for extensive retraining and precise data collection, as well as adapting to new threats. Nevertheless, there are still gaps in the field that need to be filled, and more research and development in the areas of model training and assessment techniques

8 Metadata

8.1 Reflection

The work for the project was split up pretty evenly among the group. From coding to writing the paper to presentation, the work was split up evenly. Overall, the two of us worked very well together.

8.2 Acknowledgements

The project was completed for CS6501/ECE6502 Network Security and Privacy at the University of Virginia.

8.3 Code Repository

The code/data of the project can be found at:

`https://github.com/andrewsongj/fraud-detector`

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael

Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Azal Alhowaide. Iot dataset for intrusion detection systems (ids), 2022. Accessed: 2024-05-03.

[3] Nathaniel Bastian, David Bierbrauer, Morgan McKenzie, and Emily Nack. Aci iot network traffic dataset 2023, 2023. Accessed: 2024-05-03.

[4] CICIDS2017 Dataset. Cicids2017. `https://www.kaggle.com/datasets/cicdataset/cicids2017/data`. Accessed: 2024-05-03.

[5] Cloudflare, Inc. What is the mirai botnet? — definition ddos impact — cloudflare. Accessed: 2024-05-03.

[6] Artur Marzano, David Alexander, Osvaldo Fonseca, Elverton Fazzion, Cristine Hoepers, Klaus Steding-Jessen, Marcelo H. P. C. Chaves, Ítalo Cunha, Dorgival Guedes, and Wagner Meira. The evolution of bashlite and mirai iot botnets. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00813–00818, 2018.

[7] Emily Nack. Aci iot network traffic dataset 2023, 2023. Accessed: 2024-05-03.

[8] Rachana Sharma and Inderjit Kaur. A comparison of feature engineering and classification approaches for malware detection. In *Proceedings of the 15th International Conference on Security and Cryptography - Volume 1: SECRYPT,*, pages 591–599. SciTePress, 2018.

[9] CNET Staff. A record $12.5 billion lost to internet crime in 2023. 2023. Accessed: 2024-05-03.

[10] Google Cloud Staff. 2023 zero-day trends, 2024. Accessed: 2024-05-03.

[11] Okta Staff. How cybercriminals are using gen ai to scale their scams, 2024. Accessed: 2024-05-03.

[12] Keras Team. Transfer learning guide, 2024.

[13] Ly Vũ, Quang Uy Nguyen, Diep Nguyen, Hoang Dinh Thai, and Eryk Dutkiewicz. Deep transfer learning for iot attack detection. *IEEE Access*, PP:1–1, 06 2020.

[14] J. Zhao, S. Shetty, J. Pan, et al. Transfer learning for detecting unknown network attacks. *EURASIP Journal on Information Security*, 2019(1), 2019.