International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.
© 2026 Scholar AI LLC.
https://ijpsat.org/
Vol. 56 No. 2 April 2026, pp. 151-162

# *Transfer Learning Technique: An Approach For Network Intrusion Detection*

Umejuru Daniel[1], Anthony Vivian Onyinyechi [2]

[1, 2]Department of Computer Science, University of Port Harcourt, Choba, Nigeria
E-mails: daniel_umejuru@uniport.edu.ng, vivian.anthony@uniport.edu.ng
[1]https://orcid.orcid.org/0009-0007-9843-2248
Corresponding author: Umejuru Daniel. E-mail: daniel_umejuru@uniport.edu.ng

**Abstract: Transfer learning (TL) is a machine learning technique that applies knowledge gained from solving one problem to solve a different but related problem. TL reduces the effort and time required to solve a problem by using a pre-trained model on a similar problem. Recently, network intrusion detection systems have incorporated both traditional and deep learning techniques. The goal of these systems is to detect network attacks and violations of rules. Although machine learning introduces novel attack surfaces that fascinate researchers, it also challenges the development of novel models, input definitions, and ML-IDS training. This research investigates the application of transfer learning as a technique for network intrusion detection. The proposed IDS system is capable of detecting multiple threat levels in the networking field. The system uses two pre-trained convolutional neural networks (CNNs) from the perspective of knowledge transfer, combining their predictions into a single model. The pre-trained CNNs on a related problem are adapted for a new but similar problem. The models were trained and tested using transfer learning to detect both common and uncommon attack patterns. The NSL-KDD dataset from Kaggle was employed to test the network intrusion detection system. The experimental results demonstrate that our TL technique has a prediction accuracy of 96.52%, which is a remarkable level of efficiency as expected. We therefore recommend the application of transfer learning, which has greatly contributed to the development of deep learning and is essential in developing efficient network intrusion detection systems.**

**Keywords: Intrusion, Detection, Attack, Transfer learning, Network**

## 1.0 Introduction

The use of artificial intelligence to help solve problems with massive data sets and the Internet of Things has continued to make our lives better, despite our increasing use of the World Wide Web (Wang et al., 2023). An intrusion detection system, if used as a shield, will not only protect the networking environment but also serve as a preventative measure. Networking has a lot of information, and it is supposed to be protected using all available means; however, there is nothing that is 100% safe, and there is always something new as technology continues to advance. It is due to this fact that intrusion detection systems have become necessary in information technology (Solanki et al., 2020). Identifying abnormal behavior on the network is a major challenge with intrusion detection systems, and it is becoming ever more critical as we increasingly use laptops and mobile phones (Singh et al., 2022). As our daily activities continue to move to the World Wide Web, security challenges have never been as complex as they are, especially during a pandemic (Abayomi-Alli et al., 2025).

For the detection of unusual activities on a computer or within a network, there is a special tool known as a network intrusion detection system. There are various types of IDSs, and they are used for the detection, notification, or identification of intruders trying for illegal access to the network. With the advancement of technology, an IDS is capable of learning and observing illegal access to the networking infrastructure (Singh et al., 2022). The characteristics for data collection and the patterns of the collected data, as well as the most appropriate ML model, also vary depending on the product. Based on the study by Surantha N. and Wicaksono (2020), the hierarchy of intrusion detection systems is as follows: (a) the features that are best represented for the detection of various levels of attack severity; (b) the types of data that are best represented for the detection of particular types of attack; (c) the ML methods that are most appropriate for the particular types of data; (d) the role of The ML technologies in the enhancement of the capabilities of an IDS in particular ways. Intrusion detection is the global framework for the information society and enables the provision of advanced services based on interoperable information and communications technologies (Vadhil et al., 2024). In other words, an intrusion detection system is essentially put in place to detect an intruder who is attempting to access a system unauthorized. From an IT perspective, it is a system that detects unauthorized access into a networked system or environment. In most cases, an IDS does not have any preventive measures built into it; it is more of a system that alerts you when something is wrong. Today's networks are carrying a lot of information and need to be well protected. No system is completely secure, and that is why intrusion detection systems are necessary in the information technology landscape.

Wireless Sensor Networks (WSNs) have been rising steadily and are considered one of the most promising and challenging engineering research fields. WSNs are used as a tool for monitoring and observing different physical and environmental phenomena, such as oceans, wildlife, earthquakes, pollution, wildfires, water quality, etc. WSNs can also be used as a tool for gathering data on human activities, such as health care, machine efficiency in machine manufacturing, building safety, military surveillance and reconnaissance, highway traffic, etc. (Butun, 2023).

As science and technology advance, security is one area of interest, and security-based protocol designers for computer networks are focused on enhancing security without compromising usability. There are security measures and protocols to be followed for better security against unauthorized access to computer networks. These security protocols are constantly changing with advances in science and technology, thus providing new and useful network scanning, mapping, searching, and analyzing technologies.

There have been many inventions and developments in security-based computer networking with regard to information sharing and gathering. There are many kinds of software available to tackle security issues on computer networks, such as malware protection, antivirus, Trojans, etc., and more technical ones like scanners, Network Mappers, Wireshark, etc., and even USB scanners and monitors. Unix is a welcome addition to computer networking for data sharing and gathering, with more than 600 security tools and capabilities built into the Unix operating system, which can be manually activated for security (Diogenes et al., 2023).

Networking environments are being threatened by users who are launching attacks on corporate and individual computer networks with malicious software and coding with an intention to steal information, disrupt operations, hijack networks, or even upload information for blackmailing the owners of such computer networks (Usha D. and Suganthi S., 2023).

**2.0 Overview of Transfer learning (TL)**

Transfer learning (TL) is a new paradigm that enables a model to tap into existing knowledge from related tasks, categories, or even a pre-trained model. Transfer learning, as a concept of machine learning, "involves training a model for a specific task and then using it as a starting point to learn another, related task" (Neog, et al., 2020). Typically, representations have been trained from scratch for a specific task. However, using transfer learning, a model can utilize its existing knowledge to improve performance for a new task.

**2.1.1 Types of Transfer Learning**

Transfer learning has three types: Inductive Transfer Learning, Transductive Transfer Learning, and Unsupervised Transfer Learning (Hosna et al., 2022).

**a) Inductive Transfer Learning (ITL**): This is the most widely used type of transfer learning. An algorithm is initially pre-trained and then fine-tuned for the second task by using the learned features for the first task (Tan et al., 2024).

**b) Transductive Transfer Learning (TTL):** In the case of TTL, there is a relationship between the source and target tasks, and the model is applied for prediction with the target task without any fine-tuning. This is because of the assumption that the two domains are similar.

**c) Unsupervised Transfer Learning (UTL):** It is an unsupervised model where the structure is learned from the source task and can be applied to the target task. This is particularly helpful when there is little information about the target task (Tsiakmaki et al., 2020).

### 2.1.2 TL and conventional Machine Learning (ML)

Usually, traditional machine learning starts from scratch, depending on feature engineering and optimization to guide the learning process. On the other hand, transfer learning depends on previous experiences and thus can be more data-efficient and speed up the convergence process (Tan et al., 2024).

## 2.2 Theoretical Foundations

Theoretical bases of transfer learning are founded on domain adaptation. This idea is concerned with the issue of training on a certain domain (on a particular data set) and then using the acquired knowledge on another domain (on another data set). There are a number of mathematical theories, such as domain adaptation theory and representation learning, which offer support for the idea of transfer learning, among other theories that adhere to a general principle of using knowledge from another domain to improve on a new domain

### (a). Feature Extraction

In feature extraction, the knowledge obtained from the source task is employed to select the most significant information from the data. The extracted information is then used to feed the model for the target task. The most commonly used techniques for feature extraction include Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), etc.

### (b). Fine-tuning

Fine-tuning is a technique where the pre-trained model is fine-tuned to fit the target task. In most cases, the initial layers of the network, which identify the general features, remain unchanged, while the later layers of the network are fine-tuned to fit the target task. This technique is used in computer vision and natural language processing.

### (c). Domain Adaptation

The domain adaptation techniques aim to bring the source and target domains closer to one another. The techniques employed for domain adaptation can be domain-specific batch normalization, adversarial domain adaptation networks, etc.

### (d). Pre-trained Models

Pre-trained models, such as those developed by OpenAI, including GPT and BERT, have been trained using enormous amounts of data to solve tasks such as natural language understanding. These models can be used for a variety of tasks without having to be trained afresh.

### (e). Multi-task Learning

Multi-task learning is a type of transfer learning, as described by Hosna et al. (2022). It has been applied as a method for training a model to learn multiple tasks at the same time. The main idea that this technique is based on is that a model can be made more efficient at performing tasks by learning from other tasks as well. This technique has been employed to develop many effective models.

**(f). Self-supervised Learning**

Another type of transfer learning, called self-supervised learning, has also been proposed. It has been applied as a method for training a model on a set of data without using any labels from outside sources. The labels have been generated from the data itself, as proposed by (Kim et al. 2018).

**Steps required in building transfer learning in ANN model:**

A machine learning procedure called transfer learning (TL) uses a pre-trained model that can be adapted for a new task. In other words, transfer learning uses the knowledge that a model has gained from solving a particular problem to solve a new problem.

## 3.0 Methodology

The study will employ the Object-Oriented Analysis and Design Methodology (OOADM). The OOADM seeks to analyze, design, and develop a new system as a set of interrelated classes and objects. The OOADM approach is adopted owing to its efficiency, effectiveness, reliability, and rapid development of a new system.

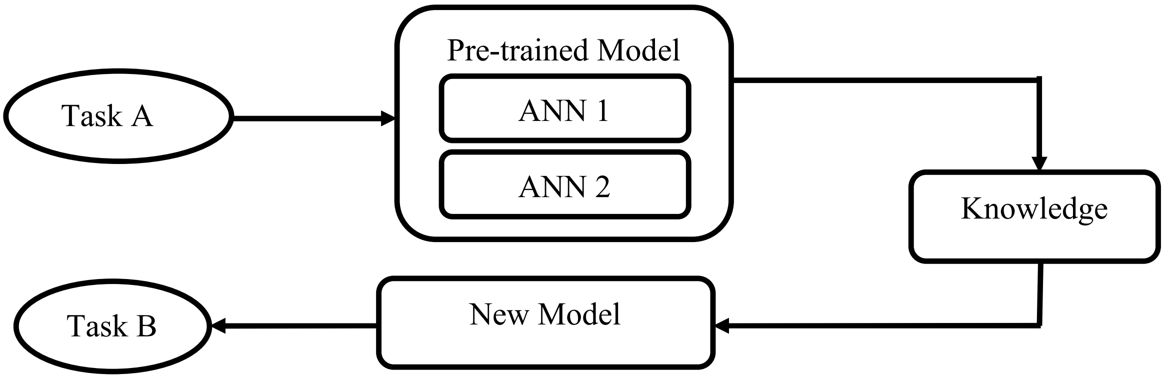


**Figure 3.1**: Transfer Learning **(Source**: GeeksforGeeks)

The low-level details learned during the execution of task A can be useful during the training of the model for task B. Transfer learning helps the model to leverage the knowledge already gained from task A while executing the subsequent task.

**Datasets:** The research utilized a dataset collected from Kaggle, and IoT Intrusion Detection System (IDS) dataset stored in CSV/Excel format. The dataset selected for this research has a vast scope for developing the proposed model, consisting of approximately 6,763 test samples and 15,780 training samples from a total of 22,543 items. For this research, the dataset used is the IoT IDS dataset from Kaggle, consisting of a total of 22,545 entries. It consists of 12,833 IDS attacks and 9,711 normal cases, represented by 42 features such as protocol, services, number of logins, number of failed attempts, etc., as presented in the following table.

**Preprocessing**: includes feature engineering and scaling to prepare the data in such a way that the algorithm can learn and identify patterns. The collection includes an IDS attack dataset, which is in Excel/CSV format. Data preparation methods such as feature scaling were used to better understand the IDS dataset and its features.

**aset Training and testing dataset:** The dataset of the proposed system is divided into 80% for training (1437 samples) and 20% for testing (360 samples) from the total 1797 samples using train_test_split with test_size parameter set to 0.2. We employ scikit-learn and ensemble techniques to create multiple decision trees and predict results using majority voting. The model is trained on 80% of the data and tested on the remaining 20%.

Table 3.1: Database table for prediction

**Algorithm:** Transfer learning in neural networks

| STEPS | PROCEDURE |
|---|---|
| 1 | Select a pre-trained model which will act as the base for training. |
| 2 | Create the base model by instantiating architecture such as a CNN. |
| 3 | Freeze the layers of the neural network so that the pre-trained weights are not overwritten. |
| 4 | Add new layers which are trainable. |
| 5 | Train the newly added layers. |
| 6 | Fine-tune the model with the given input and output pair. |
| 7 | Stop. |

| Field_Name | Data_Type | Width | Type description |
|---|---|---|---|
| ID | int | 5 | For user id |
| Protocol | int | 2 | No. of protocol units |
| Services | int | 5 | No. of Available services |
| Login_attempts | int | 5 | No. of login attempts |
| Wrong_fragments | int | 5 | Wrong fragments in network |
| No_of_compromise | int | 5 | No. of compromised attempts |
| Failed_logins | int | 5 | No. of failed login attempts |
| Ds_Hots | int | 2 | No. of Ds_hot |
| Predictions | int | 2 | Target class |

## 4.0 Result and Discussions

The results of the transfer learning model will be presented and discussed in this chapter using heat maps, a confusion matrix, bar charts, and tables. To improve the classification accuracy of the model, various hyper-parameter settings were used in the experiment for design and implementation. The confusion matrix, ROC curve, and classification report are used to illustrate the prediction accuracy of the model.

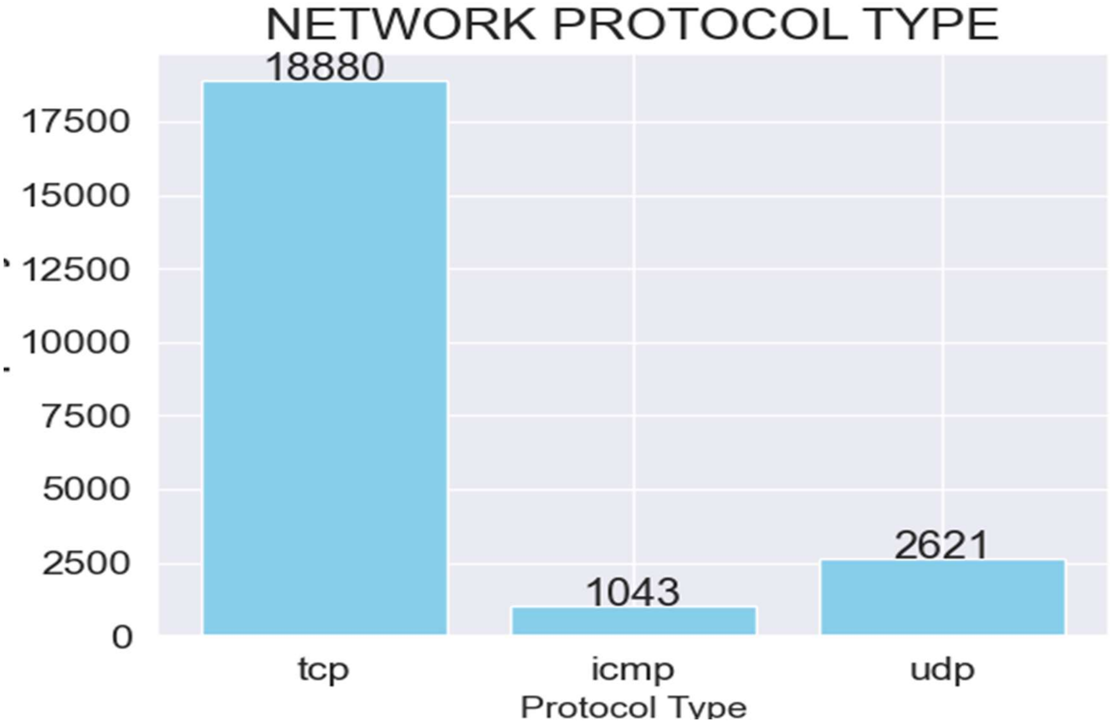
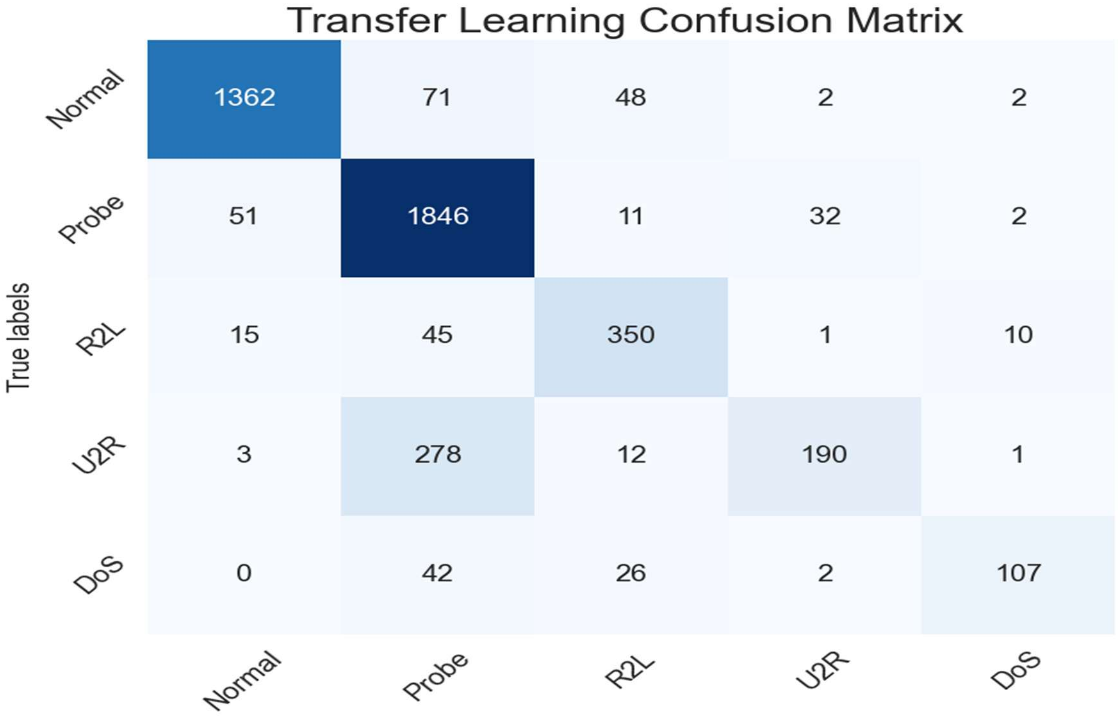


**Figure 4.1:** Protocol Types

Figure 4.1 the data indicates how frequently different protocols are used, with special interest in TCP, UDP, and ICMP. These protocols are used for different kinds of data transportation from a given source to a given destination, with faults being flagged by the use of ICMP for troubleshooting issues within a network. According to the given data, TCP is dominant with 18,880, followed by UDP with 2,621, and then ICMP with 1,043. This indicates that TCP is dominant among the protocols, followed by UDP and then ICMP. Most of the TCP is above 17,500, followed by UDP, which is slightly above 2,500, and then ICMP, which is below 2,500.

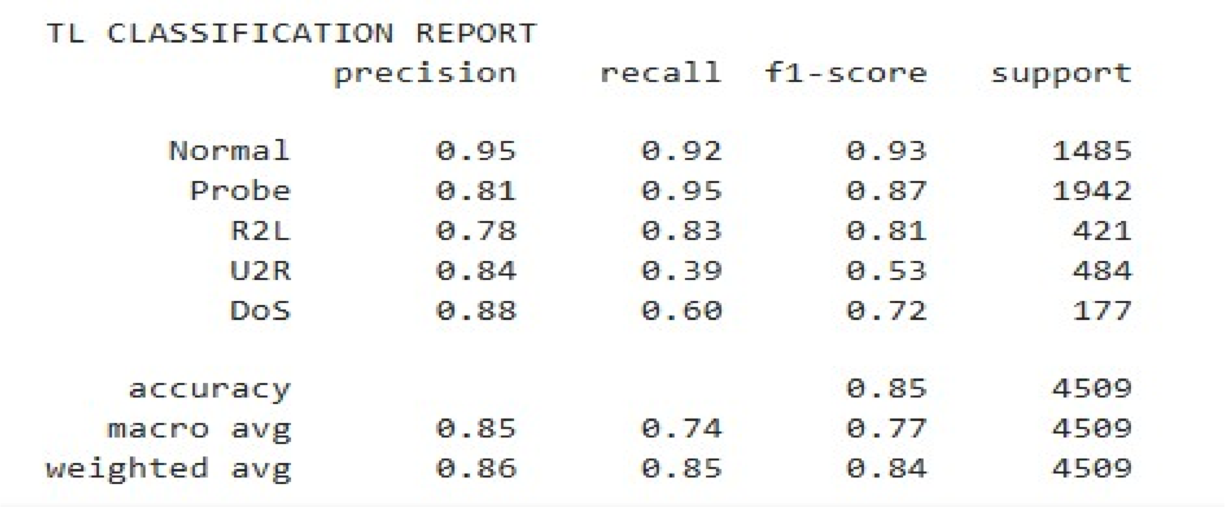


| True labels | Normal | Probe | R2L | U2R | DoS |
|---|---|---|---|---|---|
| Normal | 1362 | 71 | 48 | 2 | 2 |
| Probe | 51 | 1846 | 11 | 32 | 2 |
| R2L | 15 | 45 | 350 | 1 | 10 |
| U2R | 3 | 278 | 12 | 190 | 1 |
| DoS | 0 | 42 | 26 | 2 | 107 |

**Figure 4.2:** Transfer learning (TL) confusion matrix

Figure 4.2 hows a 4x4 matrix for the TL model, where the number of target classes used to evaluate multi-classification capabilities is indicated. Here, we are trying to evaluate how well the model's predicted results match with the actual target value, representing twenty-four distinct incompatible scenarios. According to the given data, 1362, 1846, 350, 190, and 107 represent correct classifications. On the other hand, for incorrect classifications, we have a sum of 71+48+2+2+11+32+2+1+10+1, which equals 180, and another sum of 51+15+45+3+278+12+42+26+2, which equals 474. These are instances where incorrect classifications are represented either above or below the diagonal. However, in the case of the RF classifier, we have a larger number of correct classifications compared to incorrect classifications. Moreover, we have better results for predicting the target class for normal traffic, DoS, probing, etc., in the case of IDS attacks.

**Table 4.1**: TL Classification report



```
TL CLASSIFICATION REPORT
               precision    recall  f1-score   support

      Normal        0.95      0.92      0.93      1485
       Probe        0.81      0.95      0.87      1942
         R2L        0.78      0.83      0.81       421
         U2R        0.84      0.39      0.53       484
         DoS        0.88      0.60      0.72       177

    accuracy                            0.85      4509
   macro avg        0.85      0.74      0.77      4509
weighted avg        0.86      0.85      0.84      4509
```

Table 4.1 is the classification report for TL, which shows the precision, recall, and F1-score for the IDS attack categories. The normal (benign) class shows the following: accuracy of 0.95, recall of 0.92, and F1-score of 0.93. The attack types show the following: for Probe, the value is 0.81, for R2L it is 0.78, for U2R it is 0.84, and for DoS, the value is 0.88 for all the metrics: precision, recall, and F1-score. This shows improvement, as indicated by the values in the classification report. Macro-average means all the classes are treated equally, weighted-average means the classes are scaled proportionally, and the micro-average means all the samples are treated equally.trics.

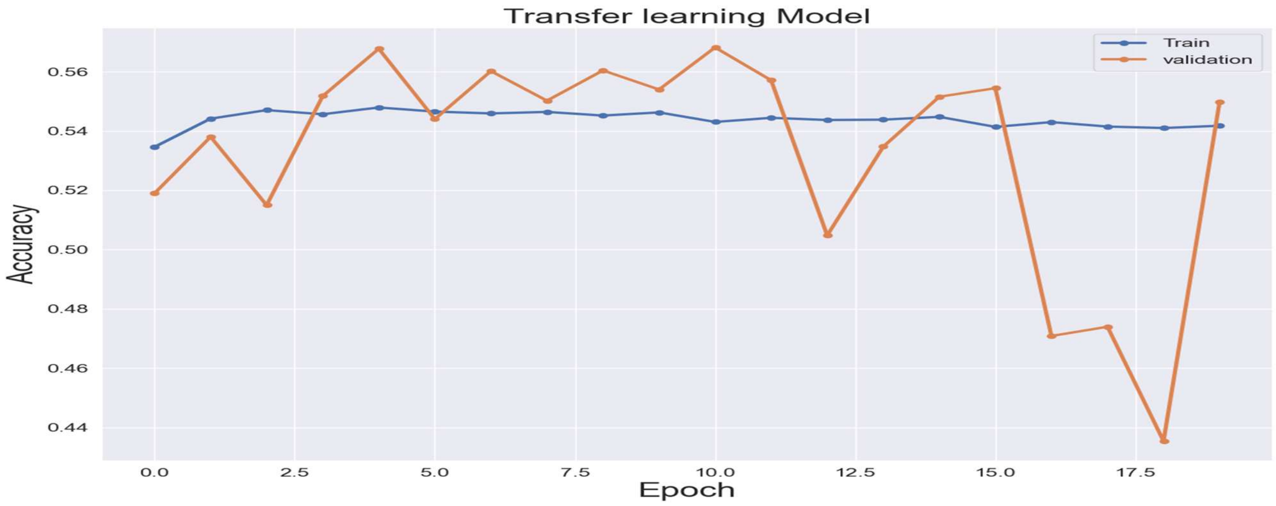


**Figure 4.3:** Transfer learning (TL) accuracy plot

Figure 4.3 the plot explains how the training accuracy and validation accuracy of the transfer learning-based model change with varying training cycles. The validation accuracy is lower than the training accuracy, implying that the model is more accurate on the validation set than on the training set. At the beginning of the training process, from epochs 0 to 2.5, the validation accuracy curve is slightly lower than the training accuracy curve, but as the training process continues from epochs 25 to 12, the validation accuracy curve increases together with the training loss curve.

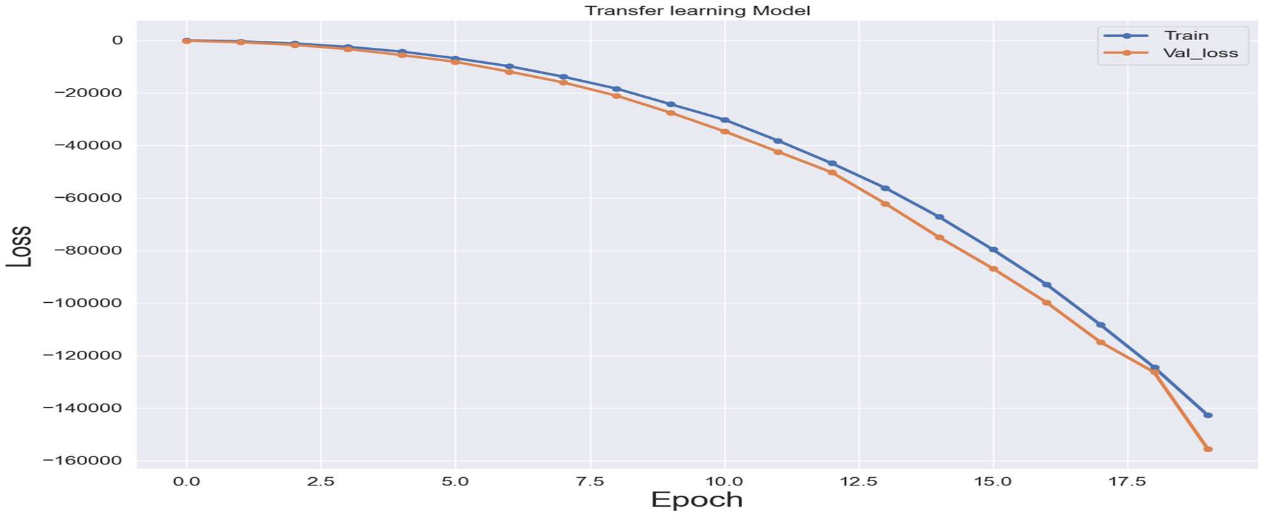


**Figure 4.4:** Transfer learning (TL) training loss

Figure 4.4 from the graph, we can understand the dependency of the accuracy of the transfer learning model and the validation loss on the initial randomly assigned model weights. The text provides us with a better understanding of the behavior of the TL model during the entire training cycle, i.e., the epochs. In the given 20+ epochs, the validation loss follows the similar pattern as the training loss. Throughout the training cycle, the training set and the validation set show an understandable correlation in terms of the error, which is the root-mean-square value.

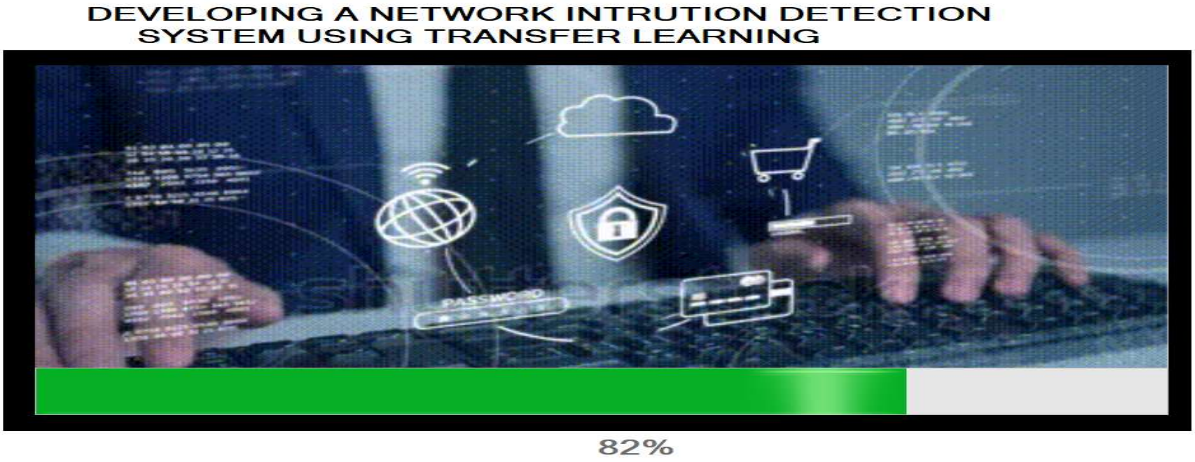
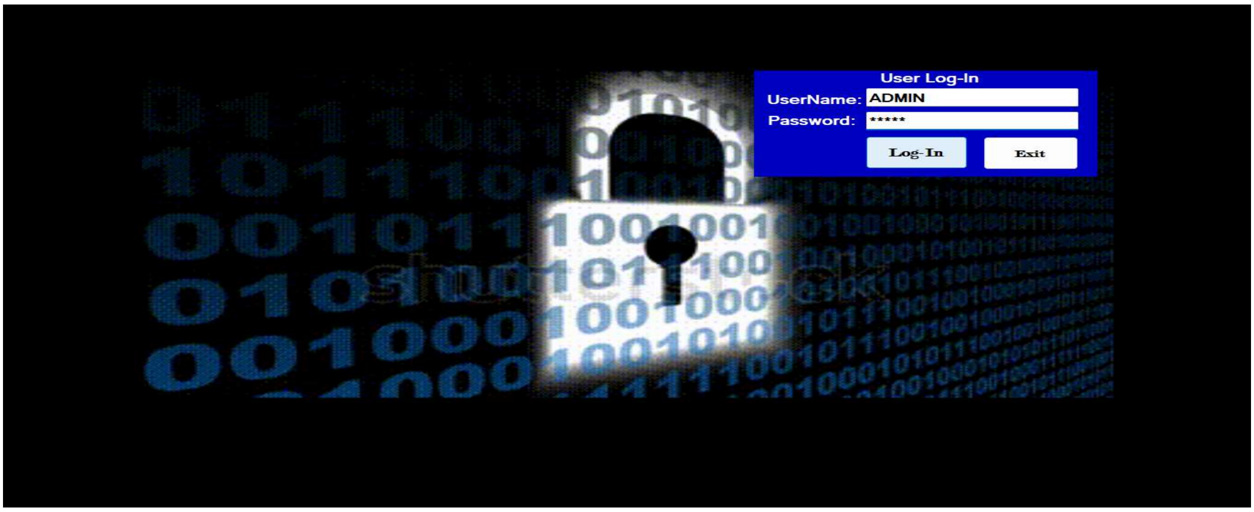

DEVELOPING A NETWORK INTRUTION DETECTION
SYSTEM USING TRANSFER LEARNING
PASSWORD
82%

**Figure 4.5**: Splash screen

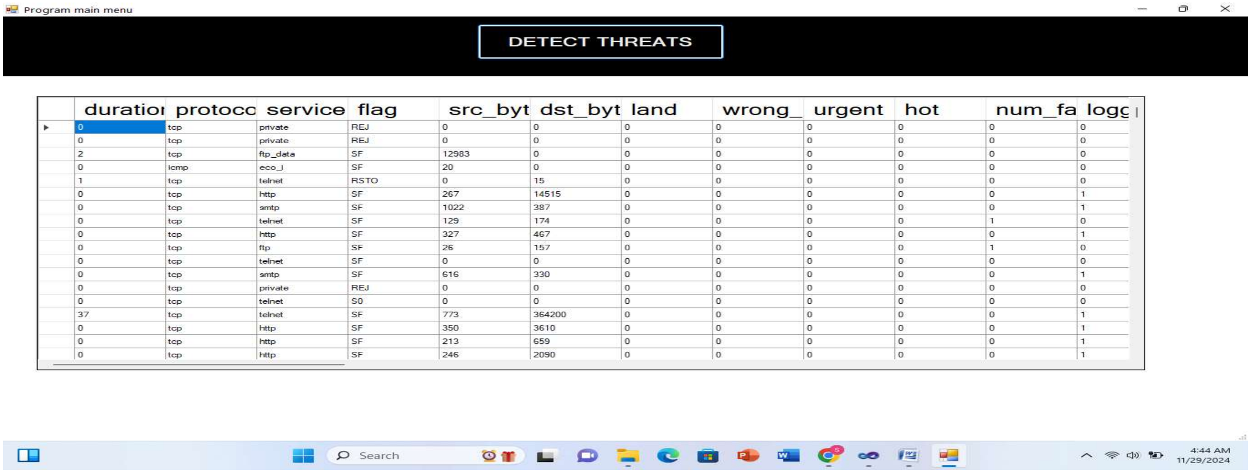
Figure 4.5 depicts the program splash screen, the first IDS program software graphic user interface, which includes a progressive bar representing the percentage of loading from 1 to 100%. The steps vary from one percent to 100%, and users must wait until the user log-in screen appears after it has reached 100%. This is the initial window that appears when a program is launched to notify the user that it has been loaded. The loading progresses as seen by the advancing bar.



UserName: ADMIN
Password: *****

**Figure 4.6**: User Login menu

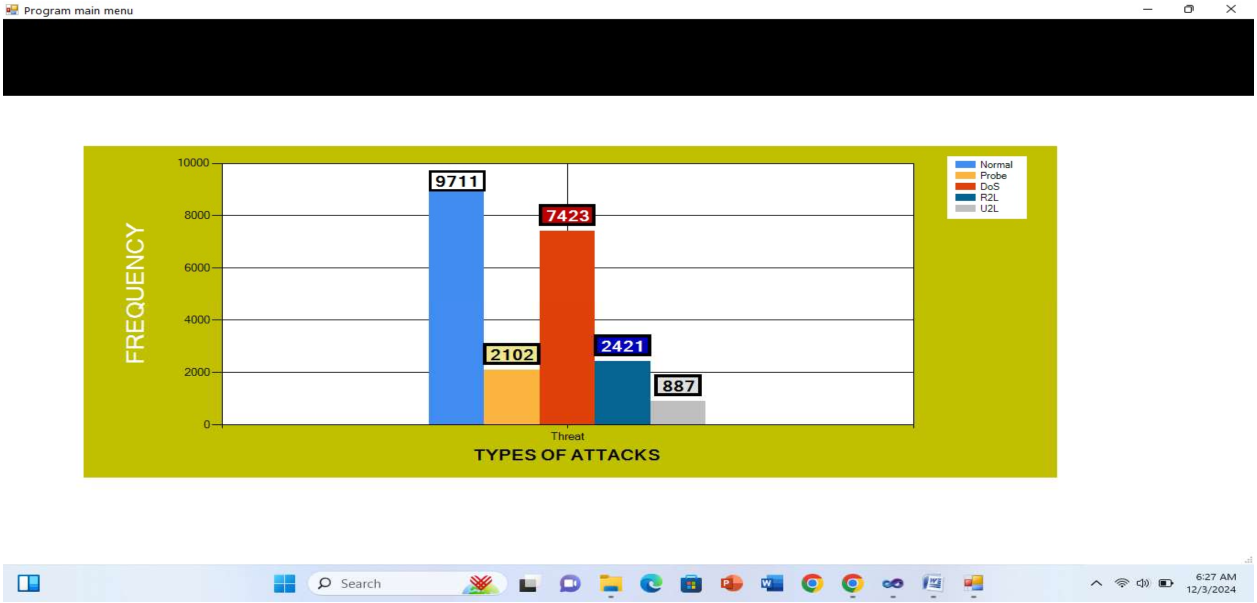
Figure 4.6 the screen above is for user login to get into the risk detection program. Here, you are required to enter your username and password. If the username and password match, then you are able to get into the program's main menu, but if they don't match, then you are kept out. If an invalid username and password are entered, then hints such as "forgot your user password" and "forgot user name" are displayed. If the password matches, then access is allowed.



| | duratio | protoco | service | flag | src_byt | dst_byt | land | wrong_ | urgent | hot | num_fa | logg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ▸ | 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 2 | tcp | ftp_data | SF | 12983 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | icmp | eco_i | SF | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | tcp | telnet | RSTO | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | tcp | http | SF | 267 | 14515 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | smtp | SF | 1022 | 387 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | telnet | SF | 129 | 174 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | tcp | http | SF | 327 | 467 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | ftp | SF | 26 | 157 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | tcp | telnet | SF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | tcp | smtp | SF | 616 | 330 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | tcp | telnet | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 37 | tcp | telnet | SF | 773 | 364200 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | http | SF | 350 | 3610 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | http | SF | 213 | 659 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | tcp | http | SF | 246 | 2090 | 0 | 0 | 0 | 0 | 0 | 1 |

**Figure 4.7**: Main menu

Figure 4.7 the image highlights the main menu of the program, which consists of three key icons: the loading of the dataset, the identification of threats, and the evaluation of the threat levels. The load IDS dataset icon allows the user to upload an Excel dataset and view the data using the data grid view..



[redacted]

**Figure 4.8**: Attack detection Page

Figure 4.8 demonstrates the chart that will be generated after a user clicks on the detect icon prior to the upload of the Excel dataset. The button for detecting represents the normal events, DoS, Probe, URR, and R2L types of attacks that are included in the dataset of the proposed system. The normal events have generated 9,711 instances, Probes have generated 2,102, DoS have generated 7,423, R2L have generated 2,421, and U2R have generated 887, as indicated in the dataset.

## 5.0 Conclusion

The effectiveness of the proposed system, based on the implementation of the Transfer Learning technique, can be concluded to be extremely efficient in terms of detecting attacks in a local and wide-area network, providing a more secured platform compared to other existing models. The effectiveness of the proposed transfer learning model, as indicated in this study, can be relied upon to address all the errors that exist in other existing models. Therefore, based on the above conclusions, we can conclusively state that the proposed system can be relied upon to address the issue of detecting IDS attacks, considering its superiority over other existing models.

## REFERENCES

[1]. Abayomi-Alli, A. A., Ikuomola, A. I., Robert, I. S. and Abayomi-Alli, O. O.(2025) An Enterprise Cloud-Based Electronic Health Records System, Journal of Computer Science and Information Technology, 2(2), 21-36.

[2]. Butun Ismail(2023)., "Prevention and Detection of Intrusions in Wireless Sensor Networks", Scholar Commons: 15

[3]. Diogenes, Y. and Ozkaya,E. (2023) Counter modern threats and employ state-of-the-art tools and techniques to protect your organization against cybercriminals, in Cybersecurity – Attack and Defense Strategies, 2nd Editio., Packt Publishing Ltd., 2-30

[4]. Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z. and Azim, M. A.(2022), Transfer learning: a friendly introduction, Hosna et al. Journal of Big Data, 1-19

[5]. Kim, D. H., & MacKinnon, T. (2018). Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks. *Clinical radiology*, *73*(5), 439-445

[6]. Neog, et al (2020). , Data Mining Techniques for (Network) Intrusion Detection Systems: 1

[7]. Solanki, S., Gupta, C. and Rai, K.(2023), A Survey on Machine Learning based Intrusion Detection System on NSLKDD Dataset, International Journal of Computer Applications, 176(20), 121- 128.

[8]. Stosic L. (2022), "Computer Security and security technmologies", Research Gate: 25.

[9]. Surantha N. and Wicaksono W. (2020), An IoT based House Intruder Detection and Alert System using Histogram of Oriented Gradients", Journal of Computer Science, 31(41), 1101-1121

[10]. Tan, Z. Jamdagni, X. He, X. Nanda, P. and Liu, R. P. (2024), A system for denial-of-service attack detection based on multivariate correlation analysis, *IEEE Transactions on Parallel and Distributed Systems*, 25, 447–456.

[11]. Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. *Applied Sciences*, *10*(6), 2145.

[12]. Vadhil, F. A., Salihi, M. L. and Nanne, M. F.(2024), Machine learning-based intrusion detection system for detecting web attacks, IAES International Journal of Artificial Intelligence (IJ-AI), 13(1), 711~721

[13]. Usha D. And Suganthi S. (2023), A Survey of Intrusion Detection System in IoT Devices, International Journal of Advanced Research (IJAR): 23, 12-23.

[14]. Wang K. and Stolfo S., (2023), "Anomalous Payload-based Network Intrusion Detection":12-23