ISSN: 2584-0495 Vol. 2, Issue 7, pp. 1050-1058

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Intrusion Detection for CyberSecurity: A Comparative Study of Machine Learning, Deep Learning and Transfer Learning Methods

Kavita Agrawal, Suresh Chittineni, P.V.G. D Prasad Reddy, K. Subhadra

Cite as: Agrawal, K., Chittineni, S., Reddy, P. V. G. D. P., & Subhadra, K. (2024). Intrusion Detection for CyberSecurity: A Comparative Study of Machine Learning, Deep Learning and Transfer Learning Methods. International Journal of Microsystems and IoT, 2(7), 1050-1058. https://doi.org/10.5281/zenodo.13332556

Published online: 17 July 2024

DOI: https://doi.org/10.5281/zenodo.13332556

International Journal of Microsystems and IoT
ISSN: 2584-0495; Vol. 2, Issue 7, pp.1050-1058; DOI: https://doi.org/10.5281/zenodo.13332556

Intrusion Detection for CyberSecurity: A Comparative Study of Machine Learning, Deep Learning and Transfer Learning Methods

Kavita Agrawal[1], Suresh Chittineni[2], P.V.G. D Prasad Reddy[3], K. Subhadra[4]

[1]Department of Computer Engineering and Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India
[2,4]Department of Computer Science and Engineering, GITAM Deemed to be University, Vishakhapatnam, India
[3]Department of Computer Science and Systems Engineering, Andhra University, Vishakhapatnam, India

ABSTRACT

With the increasing frequency and sophistication of cyber-attacks, intrusion detection has become a critical cybersecurity component to ensure the resilience and trustworthiness of modern digital systems and networks. Several machine learning and deep learning algorithms have been used. However, there is limited data on the comparative efficacy of these systems. We analyzed the usage of predefined machine learning algorithms (Logistic Regression, Decision Trees, Random Forest, Gaussian Naïve Bayes, Linear Support Vector Machine, and Gradient Boosting) and neural network centered deep learning algorithms (MLP, GRU, LSTM) and their efficiency in intrusion detection. We used the frequently used cybersecurity UNSW-NB15 dataset as our primary input for all the algorithms to test for efficacy. We then used Transfer Learning to build a more efficient model for detecting attacks using the BoT-IoT dataset (which contains a large amount of labelled data for various IoT attacks) for training and the UNSW-NB15 dataset for testing and validation. The data set consisted of around 2 million records with 49 features. By using transfer learning there was a significant increase in the percentage of various attacks detected correctly. Transfer learning appeared to be the best method for detection of the various attack categories, including known and unknown (or 'zero-day' attacks). The results need to be validated in larger data sets and ideally on real-time data to further enhance accuracy. There is a definite need to develop better intrusion detection systems that can work on large amounts of live data to keep up with the rapidly evolving cybersecurity threat landscape.

Keywords

Machine learning, Deep learning, Transfer Learning, UNSW-NB15, BoT-IoT, Zero-day attacks

1. INTRODUCTION

The world today relies on the internet and advanced technology in all domains from governmental institutions and large industries to household day to day requirements. Highly sensitive information ranging from personally identifiable information (PII), health information, intellectual property, including critical governmental and industry data systems are therefore available in these massive computer networks which need protection. Cybersecurity has therefore become an utmost priority.

Cyber-attacks on computer systems have become more frequent and sophisticated over the years driven by weaknesses in computer networks. These cyber-attacks come in many forms, such as malware, phishing, ransomware, and social engineering causing significant damage to businesses and individuals alike.

In fact, according to a recent report by the Cybersecurity Ventures organization, cybercrime alone is expected to cost the world nearly $10.5 trillion annually by 2025 primarily related to the destruction of confidential information, loss of time and money, and even the theft of intellectual property rights.

The most well-known attacks to network systems include a) Denial-of-Service(DoS) attacks(that prevent the availability of a service to a user due to an unaccountable surge in traffic or server requests), b) worms(self-replicating malware that spread ferociously, exploiting network vulnerabilities) and c) bots(allowing an attacker to gain control of a victim's system and use it as a launchpad for further malicious activity).

Intrusion detection at the earliest is the only solution to this ever-underlying threat to user security and privacy.

This is a critical cybersecurity component, that involves the process of monitoring a computer network or system for signs of any malicious activity or unauthorized access. The primary goal of intrusion detection is to try and identify and respond to security threats in real-time to prevent or minimize damage to the system, generally by initiating human intervention, generating alerts and warnings, or triggering an automated set of actions. Regardless, intrusion detection plays a crucial role in protecting critical infrastructure, sensitive user data, as well as intellectual

property, and is essential for ensuring the resilience and trustworthiness of modern digital systems and networks in a highly interconnected and dynamic environment.

Intrusion detection systems (IDS) use a variety of techniques to monitor network traffic and detect signs of suspicious activity. These systems may use signature-based detection, which compares incoming network activity to known patterns of malicious behavior, and anomaly-based detection, which identifies abnormal behavior that deviates from established norms.

Intrusion Detection Systems generally don't tend to actively remove the attacks themselves, rather, their main role lies in elevating an alarm or alert in case of any intrusion.

Over the past few years, a variety of machine learning and deep algorithms have been used to develop Intrusion Detection Systems with good accuracy and detection rates using artificial intelligence.

In this paper, we have experimented with a number of predefined Machine Learning algorithms (such as Random Forest, Decision Tree, Gaussian Naive Bayes, etc.), along with developing the neural networks for certain deep learning algorithms (GRU, LSTM, etc.) and compared their accuracies, taking a common UNSW-NB15 dataset into consideration for experimentation and measurement.

The UNSW-NB15 is a commonly used Cybersecurity dataset that is comprised of approximately 2.5 million network packets captured in a simulated network environment, with a variety of attack types. It is however, a scarcely labelled dataset, with not many fields for unknown cyber-attacks, or 'zero-day' attacks.

To circumvent this limitation, Transfer Learning methods can be used to allow for further improvement in detection. Transfer Learning is a specialized sub-concept within machine learning, that allows a base model, specifically trained for one task to be reused as the starting point for a different, related task.

Finding the ideal technique for Intrusion Detection can be a challenge requiring a fair amount of trial-and-error that could take a while to identify the best method. There is however very limited data comparing the methodologies.

In this paper, we have explored various machine learning and deep learning algorithms techniques and algorithms evaluating their performance on a commonly used dataset. Our primary objective was to identify the most timely, accurate and efficacious methodology for early intrusion detection.

There is a critical need for effective intrusion detection methods and the importance of exploring new techniques and algorithms. Our work is an attempt towards finding the best solutions for the rapidly evolving cybersecurity threat landscape.

2. LITERATURE SURVEY

2.1. Related Work

There is an increasing number of Cyberattacks in the Internet of Things (IoT) primarily because of the networks vulnerabilities. Traditionally Intrusion detection systems were either machine learning or deep learning algorithm-based models. However, often these deep learning models are unable to detect modern day cyberattacks because well-balanced datasets with labelled data may not be available in many IoT networks.

There is therefore a need to develop a detection system particularly for day-zero attacks. A transfer learning and knowledge transfer system based on convolutional neural networks model with unbalanced and unlabeled has been proposed [3].

This framework showed an accuracy of 97.89% with a low false prediction rate (FPR) of just 0.05%. At the same time the detection rates of zero-day attacks ranged from 98.85% to 100%.

The main security threats are intrusion, malware propagation, and the various attacks including distributed denial-of-service (DDoS), routing, jamming, sinkhole, sensor, replay, and mischievous sequences.

According to Zouhir Chiba et al [5] Intrusion protection systems take action based on the rule set with no human intervention. However false alarms can be generated in this system. DNN is another machine learning technique using a hybrid approach combing signature-based (SD) and anomaly detection (AD) methods for intrusion detection. This method reported a high accuracy.

Ansam Khraisat [6] classified IDS systems into Signature-based Intrusion Detection System (SIDS) and Anomaly-based Intrusion Detection System (AIDS). Datasets including DARPA 98, KDD 99, CAIDA, NSL-KDD, ISCX 2012, CICIDS 2017, Bot-IoT, ADFA-WD and ADFA-LD that are publicly available can be used to check and validate the capability of the system. The paper also highlighted that future systems should have additional self-configuration, optimization and self-healing features to reduce the number of false alarms.

On the other hand, Mansi Sahi [7] implemented a machine learning based Network Intrusion Detection (NID) system in a multi-node fog environment using a Raspberry Pi cluster on a local area network. This Pi-IDS system has been evaluated on ADFA-LD datasets. and was able to achieve a Recall of 89%in ADFA-LD with the XGBoost model. The system was suitable to prognosticate intrusion with an conclusion time 130 ms in comparison to Cloud with 735

ms, with an estimated running cost of 201 INR/ month in comparison to the Cloud cost of 2051 INR/ month.

Similarly, Laberanio Andrade-Arenas [8] has proposed an evaluative metric with the use of new model systems, such as the IoT Anomaly Detection System (AD-IoT) that uses the Random Forest (RF) machine learning algorithm to detect web attacks.

A novel ensemble Hybrid Intrusion Detection System (HIDS) by combining a C5 classifier and One Class Support Vector Machine classifier has been tried with the SIDS and AIDS systems [9]. This demonstrated higher detection rate and lower false positive rate compared to the systems individually.

Nour Moustafa [10] suggested novel detection technique to mitigate botnet attacks against DNS, HTTP, and MQTT protocols utilized in IoT networks. This Adaboost ensemble method used a combination of three machine learning techniques including decision tree, Naive Bayes (NB), and artificial neural network, to detect malicious events. The UNSW- NB15 and NIMS botnet datasets with dissembled IoT detectors' data are used to prize the proposed features and estimate the ensemble fashion. The ensemble fashion provides an advanced discovery rate and a lower false positive rate compared with each bracket fashion included in the frame and three other state- of- the- art ways.

K. V. V. N. L Sai Kiran [11] proposes machine learning models to identify attacks in IoT networks. Machine learning classifiers such as Naïve Bayes, SVM and decision tree. Here again Adaboost are built to categorize data into normal and attack classes. An IoT based platform was built and it served as a test bed to understand and perform IoT attacks on the network. This requires a good quality of data flow in the network during the attack because interception is possible only during the continuous flow of data.

Marwa Baich [12] proposes a state of the art on IoT network intrusion detection using ML techniques during the last few years. The main objective of this experiment is, first to detect whether an attack is malicious or benign (binary classification), and also to detect the type of attack, whether it is a Dos, Probe, U2R, or even R2L attack (multiclass classification). The experimental results reveal that the Decision Tree gave the best performance with an accuracy of 99.26% and a minimum prediction time of 0.4 seconds.

Naveen Saran [13] introduced an Intrusion Detection System (IDS) for detection of multi-class intrusion attacks in IOT networks. using the Intrusion Detection System dataset (MQTT-IoT-IDS2020. The overall accuracy was 97.76%, 97.80%, 97.58%, 99.98%, 99.98%, and 97.58% using the various classifiers including k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), Decision Tree (DT) and Stochastic Gradient Descent (SGD) respectively.

Solaiman Kabir [14] proposes a Convolutional Neural Network model with mish activation function and Ranger optimizer that reaches a higher degree of precision compared to previous Deep Learning models and traditional CNN models that utilize ReLU activation function and Adam optimizer. CIC-IDS-2018 dataset has been used for testing which comprises six different varieties of attacks. The model reaches an accuracy of 98.9%.

Aimin Yang [15] puts forward the LM-BP neural network model. This algorithm allows a fast optimization speed which enables optimization of the weight threshold of traditional BP neural networks. This in turn results in higher detection rate together with lower false alarm rate than the traditional BP and PSO-BP neural network models.

Arwa Aldweesh [16] provides a novel fine-grained taxonomy that categorizes the current state-of-the-art deep learning based IDSs with respect to different facets, including input data, detection, deployment, and evaluation strategies.

Geethapriya Thamilarasu [17] used deep learning algorithms to detect and intercept unwanted intrusions in IoT networks. The system developed provides security as a service at the same time allowing interoperability between various communication protocols used in IoT. This detection framework used both real-network traces for proof of concept as well as simulation for providing evidence of its scalability. The average precision rate was a high 95% against the various attack scenarios, including blackhole attack, opportunistic service attack, DDoS attack, sinkhole, and wormhole attacks.

Muder Almiani [18] presented an artificially full-automated intrusion detection system for Fog security against cyberattacks. The proposed model uses multi-layered recurrent neural networks designed to be implemented for Fog computing security that is very close to the end-users and IoT devices. The model shows high sensitivity to DoS attacks that represent one of the prominent attacks thwart the development of IoT network besides detecting other types of attacks' categories such as Probe, R2L, and U2R in a competitive computational overhead as each record requires 66 µsec on average to be processed. Thus, the proposed model is capable of properly and efficiently working in real time environments.

Ahmad S. Almogren [19] proposes an approach to detect intrusive activities quickly and accurately in the EoT network, to realize the full potential of the IoT. It proposes a deep belief network (DBN) based on an advanced intrusion detection approach. The UNSW-NB15 dataset has been used to test the proposed approach. The detection performance rate of the proposed DBN model is compared with other methods such as ANN and SVM. The proposed approach outperformed both in terms of accuracy.

Monika Vishwakarma [20] proposed a unique real-time intrusion detection system to identify malicious activity in networks. Newly developed benchmark Netflow-based dataset was used to train the model which has 20 different types of networking attacks. A packet capturing and

detecting algorithm for real-time attack detection was proposed.

Brooke Lampe [21] provides a comprehensive overview of deep learning based IDSs in automotive networks. It assembles various deep learning schemes, categorizes them according to their topologies and techniques, and highlights their distinct contributions. It analyzes each scheme's evaluation in terms of datasets, attack types, and metrics. It summarizes the results of the schemes and assesses the advantages and disadvantages of different deep learning architectures. Deep learning intrusion detection systems are capable of an amazing depth and breadth of analysis, and they can learn and develop alongside novel attacks. For the purposes of future proofing, deep learning is a promising direction for the automotive intrusion detection system.

Bhukya Madhu [22] proposed a Device-based Intrusion Detection System (DIDS) which incorporated the prediction of unknown attacks to handle the computational overhead in large networks and increase the throughput with a low false alarm rate. The proposed algorithm has been evaluated with standard algorithms, and the results show that it detects attacks earlier than standard algorithms. The computational time has also been reduced, and 99% of accuracy has been achieved in detecting the attacks.

Chunhua Zhao [23] proposes an integrated model of LCNNE based on transfer learning, aiming at solving the acquisition problems of wear particle data of large-modulus gear teeth and few training datasets. On the wear particle dataset, the model achieves the accuracy rate of 99.63%.

Selim Yılmaz [24] introduced transfer learning in the context of a routing protocol for resource-constrained wireless networks known as RPL. By leveraging the experience gained from previously trained models, the proposed approach significantly reduces learning time, which is crucial for the timely deployment of devices/networks. This work is the first to apply transfer learning in IoT security to transfer knowledge for new types of attacks and new devices. Three types of attacks are present in this, single-to-single, single-to-multi, and multi-to-multi.

Abdulmonem Alshahrani [25] used transfer learning to develop a deep learning-based proxy model for evaluating candidate IDS configurations more efficiently and accurately. This approach leverages previous experience to generate high-performing deployments for newly presented networks.

Xingguo Sun [26] developed TDL-IDS, a transfer deep learning-based IDS that can work with limited labeled data items. The proposed approach first trains a model on the source domain using LSTM and then leverages transfer learning to continue the training process on the target domain. NSL-KDD was used as the source domain, and AWID was used as the target domain during the evaluation.

Alwyn Mathew [27] achieved an accuracy of 95.3% in object detection by utilizing a pre-trained Google's inception model and feeding the transferred information to multiple fully connected layers with dropouts. This work demonstrates the potential of deep CNNs in the field of object detection.

2.2. Background

The idea behind the paper is to try to understand what the best approach is to intrusion detection, and which method gives the highest accuracy.

a) Machine Learning Algorithms

The ML algorithms tried and used in this project include:

i) Logistic Regression:

Logistic Regression is a classification algorithm that falls under the category of supervised machine learning techniques. It is a statistical method that can be used for predicting binary outcomes (0 or 1, true or false, success or failure), by estimating the probability of the output based on one or more predictor variables, and then assigning the input to the output category that has the highest probability. Mathematically, the probability of finding a binary outcome is calculated by using the formula below:

$$p = 1 / (1 + e^{(-z)})$$

and

$$z = b0 + b1x1 + b2x2 + ... + bnxn$$

The projected probability of a positive outcome (p) is determined by using the natural logarithm base (e), a linear combination of input variables (x1 to xn), and coefficients (b0 to bn).

To reduce the discrepancy between the calculated probabilities and the actual values, the coefficients are estimated based on training data. The coefficients (b1 to bn) describe the effects of each input variable (x1 to xn) on the projected probability, while the intercept (b0) represents the calculation's beginning position.

ii) Decision Tree:

A decision tree works similar to the way a flowchart works. An internal node in a decision tree corresponds to a feature test, and a branch shows the result of that test. A leaf node represents an anticipated class label.

Essentially, decision trees partition the feature space into different regions that correspond to different classes.

This algorithm learns a sequence of tests that allow the correct classification of the input data by dividing the available data into smaller subsets, based on the feature that best separates the class labels until a certain criterion is

satisfied. The decision tree can be easily visualized and interpreted but may suffer from overfitting and lack of generalization.

iii) Random Forest:

The Random Forest algorithm works by combining several decision trees to make decisions. Each tree is first trained on a subset of the data and uses a random selection of features to split the data at each node. The final prediction is made by taking a majority vote of predictions from all the individual trees. It also has the capacity to handle large datasets containing multiple features.

iv) Gaussian Naive Bayes:

The Gaussian Naive Bayes algorithm is derived from the Naive Bayes probabilistic algorithm generally used for classification. It uses Gaussian distribution to find the standard deviation and mean of the provided data. It uses Bayes' theorem and probability to predict the class of a given data point. It assumes that each feature in the data follows a Gaussian distribution and that the features are independent of each other. The probability of a given data point belonging to any each class is then calculated the data point is assigned to the class with the highest probability. The general formula for Gaussian Naive Bayes is:

v) Linear SVC:

The Linear Support Vector Classification (Linear SVC) is an SVM (Support Vector Machine) algorithm commonly employed for classification purposes. The algorithm works by determining the ideal linear boundary that may efficiently divide data into multiple classes depending on their attributes. By reducing the distance between the decision border and the closest data points (also referred to as support vectors), this goal is achieved.

This algorithm is particularly useful when we have to sort high dimensional data with multiple features.
Mathematically, it solves the following optimization problem:

minimize 1/2 ||w||^2 + C ∑(1<=i<=n) ξi
subject to yi(w^T xi + b) ≥ 1 − ξi and ξi ≥ 0 for all i,

where w represents weight vector, b as the bias, C as the regularization parameter, xi as the feature vector for the ith data point, yi is the corresponding class label (+1 or -1), and ξi is the slack variable that allows for some misclassifications. While minimising the classification error, the objective function seeks to maximise the gap between the classes.

vi) Gradient Boosting Classifier:

Gradient Boosting Classifier is a machine learning algorithm that is ensemble based and joins several decision trees to make a more accurate prediction. Each decision tree is collected in a sequential manner, with each subsequent tree being built in such a manner so as to rectify the errors encountered in the previous tree, until a desired level of accuracy is finally attained. Gradient Boosting Classifier is known for its high accuracy, therefore marking its place in industries such as finance, healthcare, and e-commerce. However, it is generally more sensitive to outliers and is preferred for dealing with low-dimensional datasets with complex inter-feature relationships.
Gradient boosting combines several weak models into a stronger predictive model; the basic formula being:
F(x) = b0 + b1h1(x) + b2h2(x) + ... + bnhn(x)
where F(x) represents the predicted target variable for the input features x, b0 is the bias term, bi is the weight assigned to the ith decision tree, and hi(x) is the ith decision tree, which is trained to predict the residuals of the previous tree.

b) Deep Learning Algorithms

Moving on to the pre-defined deep learning algorithms with neural networks that were used to further improve accuracy:

i) MLP:

Multilayer Perceptron (MLP), also known as the basic "plain vanilla" form of neural networks, is made up of several layers of nodes, with each node in one layer connected to every node in the next layer. In this neural network, the first layer is the input layer that takes in data. Later, intermediate "hidden layers" process the data to extract certain elements relevant to the task at hand. In the output layer, the outcome is anticipated. MLP uses back propagation during its training and adjusts the weights and biases of each node to reduce error and improve accuracy in prediction.

ii) LSTM:

Long Short-Term Memory is a Recurrent Neural Network(RNN). It comprises of an additional "memory cell" that selectively retains information for longer periods of time, while simultaneously allowing for the removal of irrelevant information.
In jobs involving such data, such as speech recognition and natural language processing, LSTM is a form of architecture that describes long-term dependencies found in data sequences.

iii) GRU:

A Gated Recurrent Unit (GRU) is an RNN that deals with the 'vanishing gradient' problem found in traditional RNNs. It is a modified version of the LSTM algorithm and has a hidden state that combines both short-term and long-term

memory, making it faster to train and more computationally efficient. GRU utilizes a gating mechanism to selectively allow information to pass through the network, thereby allowing it to retain important information over longer periods of time. It is commonly used in speech recognition, language modeling, and machine translation.

c) Transfer Learning

Transfer learning is a technique of machine learning that transfers knowledge from one task to another related task for more efficient and effective training. This strategy can be useful when there is a lack of training data for a target task since it enables the model to use information from a source task to improve its performance on the target job. The basic process flow for Transfer Learning is shown in Fig1.

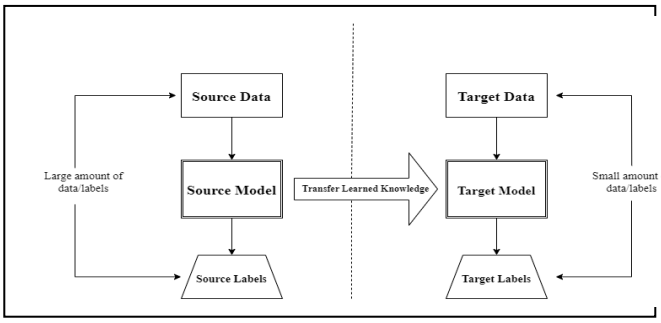

FIGURE 1. Basic Process Flow Diagram For Transfer Learning

3. METHODOLOGY

3.1 Design

Deep learning frequently employs transfer learning (TL) with pretrained models, the majority of which are based on convolutional neural networks. (CNNs). A classifier and a convolutional base for feature extraction make up the two main parts of a typical CNN. The early layers of deep learning models extract broad information, while the latter layers are focused on and biassed towards the learning task. In order to improve the learning of specialised features for a new task, TL takes advantage of this by using the general features from a pretrained model.

Stage 1: Preprocessing of source(base) dataset
Stage 2: Training the base model (CNN-Base) on base dataset.
Stage 3: Preprocessing of target(final) dataset
Stage 4: Training the Transfer Learning model on the target dataset, deriving from base model.
Stage 5: Testing the detection of attacks using target dataset.

3.2 Dataset

The primary dataset used in this comparative study is the UNSW-NB15 dataset [28].

The effectiveness of any Intrusion Detection System can be evaluated based on their ability and performance in identifying attacks. To test this, a comprehensive data set is required that has a combination of normal and abnormal behaviors. Initially, NSL-KDD and KDDCUP99 were the datasets that were used to test the effectiveness of intrusion detection systems; however, these datasets are now obsolete and outdated, containing redundant and missing data. They also lack the inclusion of several modern cyberattacks.

The dataset known as UNSW-NB15 is widely considered as a highly comprehensive resource for detecting intrusions in computer networks. It contains a variety of real-time network data.
UNSW-NB15 has about 2 million records that was detected by using the tcpdump tool to capture about 100GB of raw incoming data.
The BoT-IoT dataset [29] is a collection of traffic data that was gathered from a network of Internet of Things (IoT) devices. The dataset was created for the purpose of analyzing and detecting botnet attacks on IoT devices. It includes both benign and malicious traffic data, with the latter consisting of traffic generated by various types of botnets. The dataset contains information such as packet size, source and destination IP addresses, etc.

4. IMPLEMENTATION AND RESULTS

4.1 Standard Evaluation Metrics

To assess the performance of our different approaches for detecting cyberattacks, we employ certain metrics, including accuracy, precision, and recall. To compute these metrics, a confusion matrix is used to present the classification results. The matrix counts the number of records correctly classified as attacks (true positive or TP) and normal traffic (true negative or TN), as well as the number of records misclassified as attacks (false positive or FP) and normal traffic (false negative or FN).

Accuracy is the ratio of correct predictions to the total number of items evaluated:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

Precision is the ratio of values that are correctly classified as belonging to a particular class out of the total items predicted to belong to that class.

$$\text{Precision} = TP/(TP+FP)$$

Recall is the ratio of values that are correctly classified as belonging to a particular class out of the total items that truly belong to that class.

Recall = TP/(TP+FN)

4.2 Results

a) Machine Learning

As mentioned, the accuracies, precision, recall values and the time taken are recorded for each machine learning algorithm used.
The values attained can be depicted as follows in Fig 2:

Table 1. Results of Machine Learning Algorithms

Algorithm	Accuracy %	Time taken	Precision	Recall
Random Forest Gini	92.61	255.54	0.91	0.89
Gradient Boosting	92.10	1248.93	0.91	0.87
Decision Tree	91.44	38.09	0.89	0.87
SVM	89.19	388.62	0.93	0.76
Logistic Regression	89.03	80.01	0.92	0.76
Gaussian Naïve Bayes	50.46	18.30	0.42	1.00

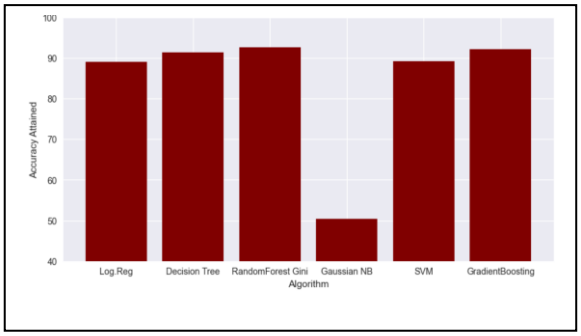

Figure 2. Comparison of Machine Learning Algorithms

It is seen that the Random Forest algorithm gives the highest accuracy among the ones measured, but it still has a relatively high execution time as shown in Table 1 .
A more accurate result is then tried for using deep learning algorithms.

b) Deep Learning

Similar to the process behind evaluating the machine learning algorithms, the accuracy, precision, recall and time values are calculated for the above used LSTM, GRU and MLP algorithms and compared, as depicted below in Table 2:

Table 2 Results of Deep Learning Algorithms

	Accuracy	Recall	Precision	time to train	time to predict	total time
MLP	96.48%	96.48%	96.49%	111.4	63	174.5
MLP (Keras)	96.19%	96.19%	96.19%	54.5	3.4	57.9
GRU (Keras)	96.53%	96.53%	96.53%	118.0	5.2	123.2
LSTM (Keras)	96.44%	96.44%	96.44%	111.4	63	174.5

Here, the highest accuracy is attained by the GRU algorithm, i.e., 96.53%, however the values between the algorithms are very close, and may slightly vary depending on the number of times the neural network models are retrained and number of epochs is changed.

After exploring deep learning algorithms to achieve higher accuracy, a transfer learning approach was applied to see if further accuracy can be attained, and an efficient model can be developed. Transfer learning has shown great promise in a number of fields, including computer vision and natural language processing, and its application in cybersecurity is on the rise.

c)Transfer Learning

Through the use of two datasets, UNSW-NB15 Test-Extra, which includes 5 types of previously unidentified assaults, and UNSW-NB15 Test, which includes both normal behavior and all 9 forms of attacks, the effectiveness of the transfer learning model is evaluated. This makes it possible to thoroughly test and validate the model. The number of identified samples, undiscovered samples, and detection rates for the five different zero-day assaults are also shown in table 3 below.

Table 3 Results of Transfer Learning approach

Traffic	Detected %	No Detected %	Detected Samples	No Detected Samples
Normal	98.33	1.66	30358	513
Analysis	100	0	622	0
Backdoor	100	0	357	0
Fuzzers	99.95	0.05	21507	10
Shellcode	99.93	0.07	1510	1
Worms	98.85	1.15	172	2

5. CONCLUSION & FUTURE SCOPE

In conclusion, intrusion detection is a crucial facet of cybersecurity and a successful method for spotting and addressing security risks instantly. Intrusion detection systems have developed to incorporate machine learning and deep learning algorithms in order to improve detection rates and accuracy in response to the increasingly sophisticated nature of cyberattacks. Transfer learning has emerged as the most effective approach in this study for enhancing the detection of unknown or zero-day threats in sparsely labelled datasets.

Building an efficient model for the detection of attacks can be beneficial in many ways, with its applications ranging from network security to fraud detection.

With the increasing complexity and frequency of cyber-attacks, organizations are constantly looking for better ways to detect and respond to threats in real-time, which could be a potential future scope of this project, where the data can be captured live and detected on any user system.
It can help automate the process of identifying and mitigating these threats, thereby reducing the workload on security teams and improving response times.

Intrusion detection will continue to be essential in safeguarding the security and privacy of people and enterprises in a highly interconnected and dynamic environment as the world becomes more dependent on technology.

REFERENCES

1. Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. Ieee Access, 7, 82512-82521, https://doi: 10.1109/ACCESS.2019.2923640.

2. Wu, P., Guo, H., & Buckland, R. (2019). A transfer learning approach for network intrusion detection. In 2019 IEEE 4th international conference on big data analytics (ICBDA) (pp. 281-285). IEEE., https://doi: 10.1109/ICBDA.2019.8713213.

3. Rodríguez, E., Valls, P., Otero, B., Costa, J. J., Verdú, J., Pajuelo, M. A., & Canal, R. (2022). Transfer-learning-based intrusion detection framework in IoT networks. Sensors, 22(15), 5621, https://doi.org/10.3390/s22155621

4. Kumar, S., Gupta, S., & Arora, S. (2021). Research trends in network-based intrusion detection systems: A review. IEEE Access, 9, 157761-157779, https:// doi: 10.1109/ACCESS.2021.3129775.

5. Chiba, Z., Abghour, N., Moussaid, K., Lifandali, O., & Kinta, R. (2022). A deep study of novel intrusion detection systems and intrusion prevention systems for Internet of Things Networks. Procedia Computer Science, 210, 94-103.. https://doi.org/10.1016/j.procs.2022.10.124.

6. Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, 1-27. https://doi.org/10.1186/s42400-021-00077-7

7. Sahi, M., Soni, M., & Auluck, N. (2021). An intrusion detection system on fog architecture. In 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) (pp. 591-596). IEEE., https://doi: 10.1109/MASS52906.2021.00084.

8. Andrade-Arenas, L., & Ramos-Romero, J. A. (2020). Analysis and prevention of IoT vulnerabilities by implementing a lightweight AD-IoT intrusion detection system model. In 2020 IEEE Congreso Bienal de Argentina (ARGENCON) (pp. 1-4). IEEE., https://doi: 10.1109/ARGENCON49523.2020.9505497.

9. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics, 8(11), 1210.. https://doi.org/10.3390/electronics8111210.

10. Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal, 6(3), 4815-4830. https://doi.org/10.1109/JIOT.2018.2871719.

11. Kiran, K. S., Devisetty, R. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a intrusion detection system for IoT environment using machine learning techniques. Procedia Computer Science, 171, 2372-2379. ,https://doi.org/10.1016/j.procs.2020.04.257.

12. Baich, M., Hamim, T., Sael, N., & Chemlal, Y. (2022). Machine Learning for IoT based networks intrusion detection: a comparative study. Procedia Computer Science, 215, 742-751. https://doi.org/10.1016/j.procs.2022.12.076.

13. Serhane, A., Hamzaoui, E. M., & Ibrahimi, K. (2023). IA Applied to IIoT Intrusion Detection: An Overview. In 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-6). IEEE. https://doi.org/10.1155/2021/7154587

14. Kabir, S., Sakib, S., Hossain, M. A., Islam, S., & Hossain, M. I. (2021). A convolutional neural network based model with improved activation function and optimizer for effective intrusion detection and classification. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 373-378). IEEE. https://doi: 10.1109/ICACITE51222.2021.9404584.

15. Yang, A., Zhuansun, Y., Liu, C., Li, J., & Zhang, C. (2019). Design of intrusion detection system for internet of things based on improved BP neural network. Ieee Access, 7, 106043-106052. https://doi: 10.1109/ACCESS.2019.2929919.

16. Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems, 189, 105124. https://doi.org/10.1016/j.knosys.2019.105124.

17. Thamilarasu, G., & Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. Sensors, 19(9), 1977. https://doi.org/10.3390/s19091977

18. Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101, 102031. https://doi.org/10.1016/j.simpat.2019.102031.

19. Almogren, A. S. (2020). Intrusion detection in Edge-of-Things computing. Journal of Parallel and Distributed Computing, 137, 259-265. https://doi.org/10.1016/j.jpdc.2019.12.008.

20. Vishwakarma, M., & Kesswani, N. (2022). DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. Decision Analytics Journal, 5, 100142., https://doi.org/10.1016/j.dajour.2022.100142.

21. Lampe, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications. Expert Systems with Applications, 119771. https://doi.org/10.1016/j.eswa.2023.119771.

22. Madhu, B., Chari, M. V. G., Vankdothu, R., Silivery, A. K., & Aerranagula, V. (2023). Intrusion detection models for IOT networks via deep learning approaches. Measurement: Sensors, 25, 100641. https://doi.org/10.1016/j.measen.2022.100641.

23. Zhao, C., Lin, Z., Tan, J., Hu, H., & Li, Q. (2022). A new transfer learning ensemble model with new training methods for gear wear particle recognition. Shock and Vibration, 2022, 1-10.. https://doi.org/10.1155/2022/3696091

24. Yılmaz, S., Aydogan, E., & Sen, S. (2021). A transfer learning approach for securing resource-constrained iot devices. IEEE Transactions on Information Forensics and Security, 16, 4405-4418. https://doi: 10.1109/TIFS.2021.3096029.

25. Alshahrani, A., & Clark, J. A. (2022). Transfer Learning Approach to Discover IDS Configurations Using Deep Neural Networks. In 2022 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI) (pp. 1-8). IEEE. https://doi: 10.1109/CCCI55352.2022.9926695.

26. Sun, X., Meng, W., Chiu, W. Y., & Lampe, B. (2022). TDL-IDS: Towards a transfer deep learning based intrusion detection system. In GLOBECOM 2022-2022 IEEE Global Communications Conference (pp. 2603-2608). IEEE. https:// doi: 10.1109/GLOBECOM48099.2022.10001267.

27. Ajayi, O. (2022). Developing Cross-Domain Intrusion Detection Systems (Doctoral dissertation, University of Maryland, Baltimore County). https://doi.org/10.1016/j.procs.2017.09.132.

28. Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

29. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems, 100, 779-796.

AUTHORS

Ms. Kavita Agrawal received her B.Tech degree in Computer Science and Engineering from UP Technical University, Lucknow India and M.Tech degree from Guru Gobind Singh Indraprastha University, Delhi , India .She is currently pursuing her Ph.D degree in Computer Science and System Engineering from Andhra University, Vishakapatnam, India. Her areas of interest are Blockchain, Cyber Security, Internet of Things and Machine learning.

Corresponding Author Email: Kavita.courses@gmail.com

Suresh Chittineni received his B.Tech degree in Computer Science and Engineering from VR Siddhartha Engineering College ,Vijayawada , India and M.Tech degree in Networking and Internet Engineering from SJCE,Mysore , India .He received his Ph.D degree in Computer Science and System Engineering from Andhra University, Vishakapatnam,India. His areas of interest are Adhoc Sensor Network,Deep Learning and Soft Computing.

P.V.G. D Prasad Reddy received his M.Tech degree from Andhra University, Vishakapatnam, India .He received his PhD degree in Computer Science and System Engineering from Andhra University, Vishakapatnam,India. His areas of interest are Security,IoT,Software Engineering and Wireless Network.

Subhadra Kompella received her M.Tech degree in Computer Science and Technology from Andhra University, Vishakapatnam, India and Ph.D degree from Jawaharlal Technological University, Hyderabad , India . Her areas of interest are Data Mining, Text Mining, Data Analytics and Machine Learning.