Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 5, 2025

INTRUSION DETECTION IN IOT THROUGH DEEP LEARNING

Muhammad Awais[*1], Irshad Ali[2], Muhammad Zulkifl Hasan[3], Syed Munam Ali[4], Muhammad Zunnurain Hussain[5]

[*1]*Department of Computer Science, Bahria University Lahore, Pakistan.*
[2]*Department of Computer Science, University of Engineering and Technology Taxila, Pakistan.*
[3]*Department of Computer Science, Bahria University Lahore, Pakistan.*
[4]*Department of Computer Science, Bahria University Lahore, Pakistan.*
[5]*Department of Computer Science, Bahria University Lahore, Pakistan*

[*1]m.awais.yousaf01@gmail.com, [2]wahirshad82@gmail.com, [3]zulkifl.hasan@ucp.edu.pk, [4]s03135201009@gmail.com, [5]zunnurain.bulc@bahria.edu.pk

DOI: https://doi.org/10.5281/zenodo.15422045

Keywords
Deep Learning, Intrusion Detection, Internet of Things.

Article History
Received on 07 March 2025
Accepted on 07 April 2025
Published on 15 May 2025

Corresponding Author: *

Abstract

Intrusion detection forms an important element to security in the realm of Internet of Things since it aids early identification of any incidence to compromise with the system. In the case of IoT contexts, intrusion detection processes are often futile due to the sheer volume of data produced by these gadgets. Hence, deep learning and several state-of-the-art and effective intrusion detection technologies that is required. The deep learning domain of machine learning applies neural networks to a huge amount of information.

It has been proven exceptionally effective in various applications such as, audio and picture recognition, natural language processing and anomaly detection. In terms of IoT security, deep learning can be employed to detect odd behavior in a device's activity and enable timely detection of potential security abnormalities. Thus, this research has great importance as it can help to minimize dangers associated with IoT gadgets and protect confidentiality and personal information of users.

Deep learning algorithms are expected to be the solution to the intrusion detection problem in the Internet of Things. In fact, a recurrent neural network (RNN) is used to recognize abnormal behavior by first training on the normal activity of large numbers of devices. As this model has been trained using an unsupervised learning method (i.e.,without labelled data), it is particularly suited to the challenges of security in the world of Internet-of-Things technology. Implementing this solution involves collecting and cleaning data from Internet of Things devices, then training the RNN model to detect abnormalities. The model is evaluated using standard measures such as accuracy, recall and F1 score.

Analyzing the results, it seems that deep learning techniques can indeed be used to identify aberrant behavior among Internet of Things (IoT) devices. It would seem then that they could play a role in hardening IoT device security and reducing attack risk.

INTRODUCTION

Deep learning based intrusion detection in IoT apart from rival products or services. The evolution of BC is driven by the advancement and is an application of sophisticated machine learning approaches towards identifying security issues pertaining to IoT devices and networks.

The use of IoT devices is on rise across different sectors such as medicine, transport, and manufacture. Such devices are commonly linked to the internet and with other gadgets making an incredibly vast net that may be prone to attack. IDS may be employed to identify and thwart intrusions into IoT devices and networks. Machine learning, deep learning especially uses artificial neural networks in processing data for learning. This means that deep learning algorithms can automatically extract features from data, which is very important when dealing with large datasets such as those sourced from different IoT devices.

For intrusion detection of IoT using deep learning [1], data gathered from different resources that include sensors, network traffic, and device logs is trained for deep learning model implementation. Subsequently, anomalies, threats, and trends in the information can be identified and detected for purposes of securing the organization's assets. It has also been found to be highly scalable since there many numbers involved in IoT devices. Secondly, it is able to learn with real time new data, making necessary changes in line with dynamic threats. Lastly, it can give out very precise outcomes with a smaller percentage of incorrect negatives and incorrect positives.

A. Problem Statement

Deep learning intrusion detection scope for IoT is extensive. Anomaly detection in Internet of Things (iot) devices using deep learning technique gives the indication of threat or invasion. As IoT devices permeate homes, offices, and even some public places, the importance of an efficient intrusion detection system cannot be understated.

The use of deep learning techniques like neural networks, CNNs, and RNNs will also help in detecting abnormal activities that might indicate security threats. Such methods could be used to track the behavior of devices, identify suspicious activity and raise an alarm when a danger is perceived.

Deep learning also helps in increasing the accuracy and enhancing the performance of intrusion detection system as they learn from previous experiences and adjust themselves to new ones. doing so will reduce the number of false positives and increase the efficiency of the overall system.

B. Scope

Deep learning for intrusion detection in IoT involves employing deep learning algorithms to find intrusion into IoT nodes or data. By using big data, deep learning algorithms have become very efficient for spotting irregularities as well as recognizing patterns in order to mitigate security attacks on IOT systems in real time.

These include smart homes, wearables, industries, or healthcare systems, comprising IoT devices. Nevertheless, this equipment is usually susceptible to various security issues due to a lack of adequate processing power and storage space.

Deep learning intrusion detection in IoT [2]will not only protect the devices and the data they collect from cyber attacks but also allow for the proper functioning of these devices and the privacy and security of their users.

Deep learning process for intrustion detection on IoT comprises of various stages. To begin with, there is the analysis of the dataset containing unintentional or strange conduct obtained from the IoT system. Using this dataset, a deep learning model, e.g., CNN or RNN is trained for recognizing normal behavioural pattern and identification of abnormal situation. Upon completion of training, the model can then be applied on the IoT system for real time monitoring of activities.

The model can thus activate a warning or implement measures to contain harm when it spots undesirable conduct. In addition, the new model can be retrained repeatedly using fresh data in making its predictions more accurate with time.

C. Beneficiaries

The beneficiaries of this project would be:

1) IoT device manufacturers: With this model being implemented into their products, manufacturers will have guaranteed secure devices which would be difficult for hackers to compromise. It will be a step towards increasing customers' trust, as well as creating the loyal base for the brand.

2) End-users: It can also be combined with security systems for reinforced invasion prevention. This will provide extra security and improved privacy for end-users.

3) Governments and regulatory bodies: Through this model, governments would be able to impose cyber security rules while enforcing set standards. It will also curb cyber attacks that disrupt such critical infrastructure.

4) Security professionals: This model is useful for security professionals in identifying possible threats and investigating breached security. Besides, it can be used to
develop relevant security policies and procedures.

D. Objectives

Deep learning for intrusion detection in IoT aims at protecting the security of IoT systems against various attacks targeting unauthorized entry. More specifically, some of the key objectives of intrusion detection in IoT through deep learning include:

1) Detecting anomalous behavior: This also permits identification of irregular behaviour using deep-learning models on IoT, and thus detection of possible breakthroughs to security.

2) Identifying and classifying threats: These models are capable of classifying various kinds of threats like malwares, DOS attacks and unautherized accesses to provide appropriate response for them.

3) Enhancing accuracy: Through deep learning algorithms, it is possible to enhance the accuracy of intrusion detection by exploiting vast and highly refined datasets containing finer details for improved performance, which may not be perceived in traditional intrusion detection systems.

4) Reducing false positives: Reducing false positives through deep learning models that teach how to distinguish normal behavior versus threat poses fewer false alarms.

5) Minimizing response time: Such deep learning models help identify incoming threats instantly enabling quick measures to counterattack any intrusion before it does damages.

6) Improving security: Deep learning based intrusion detection system is able to detect and prevent attacks on IoT networks.

2. Literature Review

2.1 Intrusion Detection

Cyber security comprises intrusion detection as one of its components, which contributes significantly towards addressing issues related to threats and attacks [3]. In this new trench war in the front line of cyber security, detection is but one link on a long chain to resist intrusion. But without continuous supervision to ensure that nobody's constantly poking through your computers and communications personnel are not running around like headless chickens, planners keep forgetting that they are facing a styles with extremely high abilities. This is a proactive method of spotting a threat as it happens which can assist in evading the destruction and leakages of information [2].

Some of the techniques used by intrusion detection systems include signature-based and anomaly-based approach whereby they identify both familiar and new cyber attacks. An organization is helped by them in increasing its security profile by quickly responding on incidence and providing protection against hacks [4].

2.2 Types of Intrusion Detection

1) Host-based Intrusion Detection (HIDS): Observes the actions that take place with specific hosts or devices in order to look for the evidence of illegal entry or unwanted conduct.

2) Network-based Intrusion Detection (NIDS): Perform a real time traffic analysis in order to detect unusual behavior and irregularity.

3) Anomaly-based Detection: They focus on identifying disparities from a normative baseline of behavior.

4) Signature-based Detection: Identifies malicious activity through use of preset templates or signatures of known threats

2.3 Intrusion Detection System (IDS)

An intrusion detection system (IDS) [5] analyses ongoing events in a network or an individual host to detect malicious patterns and behaviors. It performs by measuring current actions with established signatures, baseline, or pattern of behavior, issuing alarms for abnormal behavior.

1) Components

a) Sensor: Sensors are tasked with information gathering from either the target network or host. NIDS and HIDS sensors intercept network traffic, as well as monitor respective hosts respectively.

b) Analyzers: Sensors collect data which is analyzed by analyzers for detection of any pattern or abnormalities. The detectors rely on various detection means such as signature based detection, anomaly based detection and heuristic based detection.

c) Alerting System: The IDS will trigger and alert if any dubious activity is identified when the analyzer senses it. The alerts may also contain details regarding the type of the threat, how critical it is and the appropriate measures required.

d) Response Module: Responses are also included in some IDS that can automatically initiate certain actions. This may involve blocking specific Ip addresses and isolating compromised hosts, among other solutions.

2.4 Neural Network

Neural network is a mathematical model stimulating the architecture and activity of human brain. This is an example of a machine-learning algorithm, which can learn and make decisions using data [6]. In this regard, neural network is a complex multilayer system of interconnected artificial neurons – nodes. There are many functions that these networks undertake like pattern recognition, categorization, regression, among others.

1) Neurons: A neural network is made up of neurons. Every neuron collects one or several inputs, carries out mathematical operations over them, usually some weighted summation, implements an activation function and generates an output. Every neuron in one layer acts as an input to a neuron in another layer, so the previous results yield the new output.

*2) Layers:*A neural network has neurons arranged in layers. The three main types of layers are

a) Input Layer: The first input data goes through this layer.

b) Hidden Layers: They feed the input data through weighted connections and activation functions in these layers. Hidden layers can be contained by neural networks.

c) Output layer: The last layer processes all received information and presents te result as the finished product.

3) Weights and Bias: Every association between neurons has its own weight value signifying the intensity of this linkage. Also, every neuron involves its own bias term that enables the network to notice patterns even when every input is zero. During the learning process, weights and biases are tuned in order to maximize the neural network's performance.

4) Activation Function: This includes giving the model a nonlinear approach by using an activation function. Common activation functions include:

a) Sigmoid: Maps values from zero to one.

b) *Hyperbolic Tangent (tanh)*: Like the sigmoid but converts the inputs values on a scale of –1 to +1.

c) Rectified Linear Unit (ReLU): It gives output for positive inputs as well as zero for negative inputs, thereby causing non-linear behavior in the circuit

5) Types of Neural Network: Three types of neural network are:

a) Feedforward Neural Network (FNN) b) Recurrent Neural Network (RNN) c) Convolutional Neural Network (CNN)

2.5 Deep Learning

The advent of deep neural networks which are multilayered neural networks represents the third wave of revolutions in machine learning. "Deep" in deep learning simply refers to the depth comprising a series of connected units of artificial neurons constituting these networks [7]. The deep learning model uses hierarchical presentation of features that are learned from data in a self-organized manner.

The hidden layers in deep neural networks gradually represent more complex abstractions of the input data. The different layers capture diverse levels of details such that the network understands the complex correlations and patterns in the information. Feature hierarchies are what we call this hierarchical approach, where deep learning models can automatically discover pertinent characteristic attributes without explicit feature engineering.

Deep learning has an important advantage of doing end-toend, or joint training. As a result, deep neural networks can learn without any prior extraction of relevant features but by processing raw input data and performing the task directly. It means that deep learning models are flexible enough to use them in diverse tasks—from image classification, speech recognition to natural language processing. Backpropagation is one of the processes used in training deep neural networks [8].
While learning, the model estimates values and compares them with real data. Then it corrects its own parameters such as weigths and biases by minimizing the difference between predictions and outputs. At each stage of this iterative process the model is optimized so that it can perform well on new unknown data.

Advancement in data availability, storage, and processing capabilities also contributed to the success of deep learning. Deep learning models have been successful largely owing to their capacity to analyze larger datasets, parallels processing, and specialized hardware such as GPUs.

3 Methodology

3.1 Data Collection

The initial move when building an intrusion detection framework involves obtaining data. The obtained data will then be used for training and testing purposes of the deep learning model.A This information shall be obtained from a number of IoT devices such as sensors, cameras, among other devices which provide raw data. The information should capture typical as well as atypical behaviors

3.2 Data Processing

Preprocessing of the data collected will involve elimination of relevant information. Feature extraction and feature selection will equally be part of the preprocessing stage as it involves identifying features to consider in identifying intrusions.

3.3 Deep Learning Model Implement

We shall use a number of techniques like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks to build our deep learning model. The training of the model will involve using the pre-processed datasets, while the validation process would make use of a distinct test dataset [8].

3.4 Evaluation

These metrics include accuracy, precision, recall and F1 score, which will evaluate the deep learning model's performance. This will enable a comparison of its performance with some of other state-of-the-art intrusion detection techniques.

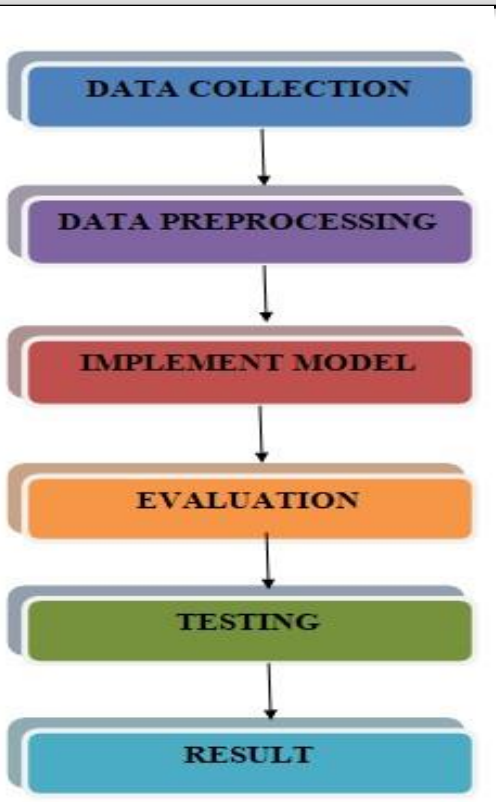

Figure 1 Methodology

3.5 Model RNN

Deep learning technique's useful tool called RNN, that can be used to detect a breach in the IoT area. RNNs in IoT Intrusion Detection with Deep Learning:

1) Sequential Data Processing: In most cases, IoT is made up of streaming of time-series data that may include sensor feeds, network packets, or system logs among other things. This suits them in being able to capture any temporal dependence or pattern there is in the streams of sequential data.

2) Anomaly Detection: It also means that RNNs can be taught to model normal behaviors of Internet of Things' (IoT) devices and networks. They can be deployed to provide for anomaly analysis by raising a flag each time they detect a departure from a previously observed pattern for intrusion analysis. Such means them efficient in recognizing innovative and unknown attacks.

3) Long Short-Term Memory (LSTM): Therefore, LSTM has become popular for IoT intrusion detection since it can exploit long-span dependencies within the data. LSTMS are good for remembering information over long durations and thus suitable identifying subtle yet complex attack methods *[9]*.

4) Real-time Detection: Optimally trained and deployed RUNNs have the ability to do intrusion detection with very high speeds compared to other methods hence they offer rapid threat mitigation in case of a dynamical IoT environment.

5) Challenges: However, there are also some limitations associated with them like scalability issues in large scale of IoT and high level of labelled data needed for training purpose. In addition, fine-tuning as well as correct adjustment of hyper-parameters is necessary for precise intrusion detection.

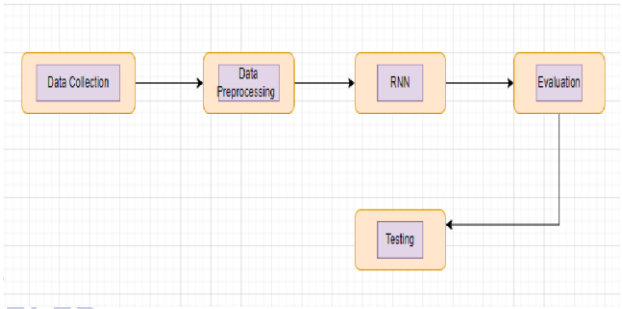

Figure 2 Methodology using RNN

3.6 Model CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks that have proven to be highly effective in areas such as image recognition and computer vision. CNNs are particularly well-suited for tasks that involve grid-like data, such as images. Here are the key concepts and components of CNNs:

1) Feature Extraction: One advantage of CNNs is that they automatically obtain hierarchical features for input data, which are already in a gridded format such as images. This is done with convolutional layers (filters learn patterns) and pooling layers (spatial dimensions are reduced).

2) Local Connectivity: Local connections in CNNs exploit local connectivity through convolutional operations, concentrating on small overlapping areas of the input. But this also helps them to record spatial hierarchies and

patterns, making them useful for activities such as image recognition.

3) Parameter Sharing: At the same time, I should remind you that CNNs employ a strategy known as parameter sharing. This cuts the number of parameters down from fully connected networks and increases generalization.

4) Translation Invariance: A convolutional layer also produces translation invariance, so the network can discern patterns at various positions. This is useful for applications such as object recognition, where the relative position of an object within a scene ought not to influence detection.

5) Hierarchical Representation: By their very nature, CNNs construct a hierarchy of features. The lower layers store simple features (such as edges), while the higher layers use them in combination to represent complex structures. The network is therefore able to learn from the data, thanks in part to this hierarchical approach.

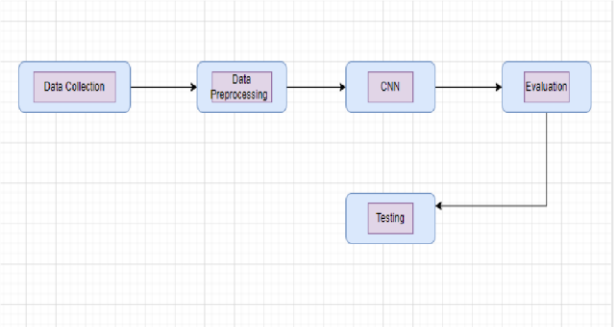

Figure 3 Methodology using CNN

3.7 FNN

One of the most common types of ANN architecture used in machine learning and/or deep learning is called feedforward neural network or MLP.

Let's explore the components and workings of an FNN in detail:

1) Input Layer: In essence, the FNN begins with an input layer, and every node corresponds to one aspect of input data. This layer's number of nodes is dictated by the dimensionality of the input data.

2) Hidden Layers: There may be one or more hidden layers between the input and output layers. Each of these hidden layers is composed of nodes (neurons) that transform the incoming input with the help of weight and bias parameters. The issue of the number of nodes in each hidden layer is one of the design choices.

3) Output Layer: The last layer of the network is responsible for generating its output. The size of the output layer is determined by the complexity and nature of a job. For example, it could be a single node for binary classification, multiple nodes for multi class classification, and even more for regression.

4) Neurons (Nodes) : The FNN consists of a number of neurons that are all linked together such that each neuron is connected to every other neuron within adjacent layers. The network learns these weights as it establishes these relationships. Activation functions are neuron's weighted sum of inputs and bias. Examples of typical activation functions that are used include sigmoid, tanh, ReLU.

5) Weights and Biases: The strength in the connecting relations of neurons also knowns as weight. These weights are adjusted during training so as to reduce the gap between the outputs. Additional parameters like biases are used by the model to counter shifts and offsets contained in the data.

6) Activation Function: Some neurons in the hidden layers, and sometimes in the output layer, are used for applying non-linear activation functions to make a system nonlinear. Such a nonlinearity allows the network to learn and represent various intricacies that exist within the data.

7) Forward Propagation: The flow of information occurs through the network forwardly during inference or training. The output layer receives input data from the hidden layers where it produces the output.

8) Loss Function: A loss function is utilized by the FNN to measure the deviation of the predicted values from the real targets. This loss should be minimized while training.

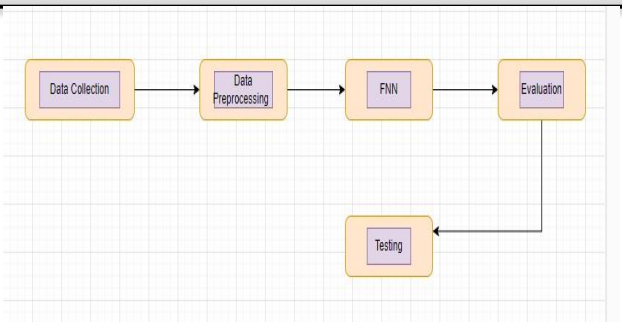

Figure 4 Methodology using FNN

3.8 Risks Involved

The use of deep learning in intrusion detection in IoT systems comes with some associated risks. Some of these risks include:

1) Data Bias: Deep learning models rely heavily on the quality and quantity of training data. A trained model may learn erroneous patterns if the training data is biased which will result into detection incorrect intrusion.

2) Adversarial attacks: In addition, adversarial attacks can trick deep learning models in such a way that even innocuous and malicious data can be misclassified as a different category.

3) Over-fitting: However, training a model with small datasets cause overfitting which implies that such models become too specialized to the specific datasets and hence poor in performance for any other data.

4) False positives and negatives: However, deep learning models can give false alarms of abnormal activity when it is normal and miss some malicious activities which eventually result into detection of abnormal activity that is not present and missed true malicious.

5) Privacy concerns: Using deep learning models demands huge quantities of data that could raise issues regarding data protection and safety.

Careful assessment of the costs and advantages associated with deep learning based intrusion detection system in IoT environment has to be conducted first and implemented proper measures should be taken to prevent possible risks.

3.9 Dataset Description

UNSW_2018_IoT_Botnet_Final_10_best_Testing dataset is utilized [10]. The data types seem to be a mix of integers, floats, and strings based on the nature of the columns. Integer types are used for counts and identifiers, floats for statistical values, and strings for categorical or textual information.

In this dataset variable our target variable is attack where it can identify whether target is happening or not.

3.10 Formula's

1) Accuracy:

Accuracy is a proportion of how many instances in the dataset have been classified correctly.

Accuracy = (TP+TN)/(TP+TN + FP + FN)

2) Precision:

Precision is the ratio of correctly identified positives among all that were predicted by the model as positives.

TP / (TP + FP), precision.

3) Recall:

Recall refers to how many correct positive classifications are recalled from all the positive cases.

TP / (TP + FN).

4) F1-score:

Harmonic mean of precision and recall (F1-score) considers both measures with equal importance. It evaluates the tradeoff of precision versus recall.

F1-score = 2 * precision x recall / (precision + recall) Where, precision and recall as already defined.

4. Implementation

4.1 Dataset Cleaning

Cleaning the dataset, or data cleaning or preprocessing, is a vital link in the chain of steps undertaken for analysis and machine learning. This process entails fixing errors, inconsistencies and inaccuracies within the data to make certain it is valid and able to be used for analysis or model training [11]. Here are some common steps and techniques for dataset cleaning:

1) Handling Missing Values: Deal with missing values in the data. This might include filling in missing values with the mean, median

or mode of its column (or some other imputation technique). If rows or columns with large numbers of missing values cannot be properly imputed then consider removing them.

2) Removing Duplicates: Remove duplicates from the data to prevent redundancy. Is there a duplicate row? Search for duplicated data, either by all columns or a few select ones (as appropriate to the context).

3) Dealing with Outliers: Select outliers for removal

VARIABLE	DESCRIPTION	DATA TYPE
PkSeqID	Packet Sequence ID	Integer
Proto	Protocol used	String - TCP/UDP
saddr	Source IP address	String - IPv4
sport	Source port number	Integer
daddr	Destination IP address	String - IPv4
dport	Destination port number	Integer
seq	Sequence number	Integer
stddev	Standard deviation	Float
N_IN_Conn_P_SrcIP	Number of inbound connections per source IP	Integer
min	Minimum value	Float
state_number	State number	Integer
mean	Mean value	Float
N_IN_Conn_P_DstIP	Number of inbound connections per destination IP	Integer
drate	Data rate	Float
srate	Session rate	Float
max	max: Maximum value (Float)	Float
attack	Indicates if it's an attack or not	String
category	Category of attack	String
subcategory	Subcategory of attack	String

and transformations to avoid skewing the analysis or machine learning model Either filter bad data, or apply a transformation that normalizes it.

1) Standardizing or Normalizing Data: Normalize (bring to a common scale) the numerical features. This is especially true for machine learning methods which require the input features to be on a certain scale.

2) Handling Inconsistent Data: Recognize and eliminate anomalies in the data. Cate-gories This may include formatting, reconciling differences or correcting spelling errors.

3) Converting Data Types: Confirm that data types are correct for every column. For instance, columns numbered 0 through x should be of numeric data type; date columns y and z are two such examples.

4) Addressing Skewed Distributions: In other words, if there are any skewed distributions of numerical features that need to be made more symmetrical, then use methods such as logarithmic transformation.

5) Handling Categorical Variables: Categorical variables can be converted into machine-readable format. This could be due to one-hot encodings, label encodings or more complex methods such as target codings.

6) Data Validation: Compare the data with predefined business constraints to ensure that it is valid.

4.2 Import the Dataset

Instead you can use the read.csv function to import your data, then open up Rattle with it in one line of code using rattle(open="datasetfile").

4.3 Explore the Dataset

Within Rattle, you can open the summary view of your data set from under the Explore tab. The Explore tab provides a summary view that lets you interactively explore and analyze your data.

1) Summary

Explore the summary of dataset.

1.1 Summary Description

In Rattle, you can use the "Summary Description" tab for a specific purpose, such as viewing summary statistics, by specifying the intended function after loading the dataset.

1.2 Summary Basic

Basic tab is use for the basic statistics for each numeric variable of the dataset.

1.3 Summary Kurtosis

In Rattle, you can access the "Summary Kurtosis" tab to view kurtosis statistics for your dataset. The "Summary Kurtosis" tab active, providing kurtosis statistics to assess the shape of the data distribution.

1.4 Kurtosis Statistics

In Rattle, you can access the "Summary Kurtosis" tab to view kurtosis statistics for your dataset. The "Summary Kurtosis" tab active, providing kurtosis statistics to assess the shape of the data distribution.

1.5 Summary Show Missing

The "Summary Show Missing" tab active, providing information about missing values in your dataset. It is used for visualize the missing data statistics of dataset.

1.6 Summary Cross-tab

The relationships between the variable that present in your dataset the cross-tab is used. The "Summary Cross Tab" tab active, allowing you to explore cross-tabulations and relationships between variables in your dataset concentration and patterns of data points, aiding in the identification of outliers and the assessment of data skewness and kurtosis.

5) **Result**

5.1. Model Accuracy

Accuracy is a measure of how well the model predicts the correct output. It is the ratio of correctly predicted instances to the total instances. In

2) Density Distribution

In Rattle, exploring the density distribution of your dataset can be conveniently done through its graphical user interface. Navigate to the "Explore" tab, where you'll find options for creating Univariate plots, including density plots and histograms. Utilize the "Univariate" and "Bivariate" of variables. The density plots, generated through kernel density estimation, provide insights into the

For sequence data or time-series prediction tasks, accuracy might be defined differently based on the specific problem.

For instance, in sequence-to-sequence tasks, accuracy might be calculated based on the correctness of each element in the predicted sequence.

5.2. Model Loss

The loss (or cost) is a measure of how well the model is performing in terms of the difference between the predicted output and the actual target. The goal during training is to minimize this loss. Different loss functions are used for different types of problems. The loss function provides a single scalar value that the optimization algorithm seeks to minimize during the training process. It is a measure of the model's error, and lower values indicate better performance.

5.3. ***Monitoring Accuracy and Loss in Training of RNN***

During the training of an RNN, accuracy and loss are typically monitored across epochs. An epoch is one pass through the entire training dataset. The model is updated iteratively to reduce the loss and improve accuracy. Plots of accuracy and loss over epochs are common visualizations to assess how well the model is learning. A model that is learning effectively will generally see decreasing loss and increasing accuracy over time *[7]*.

classification problems, where the goal is to assign a label to input data, accuracy is often used as a performance metric.

It is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

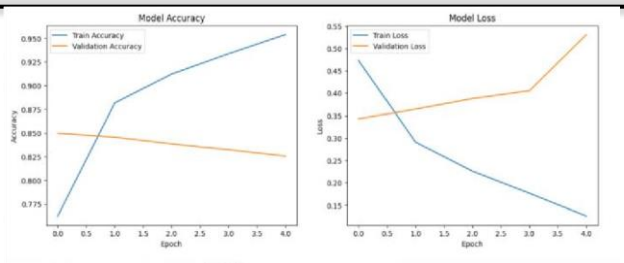

Figure 5 RNN Model Accuracy and Loss Graph

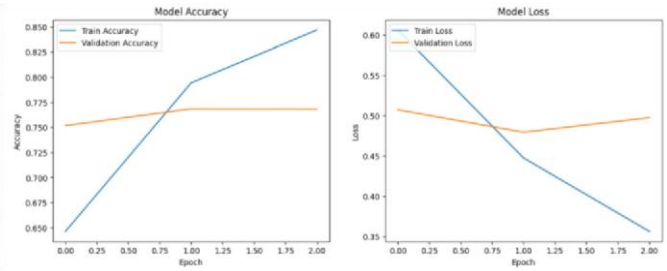

Figure 6 CNN Model Accuracy and Loss Graph

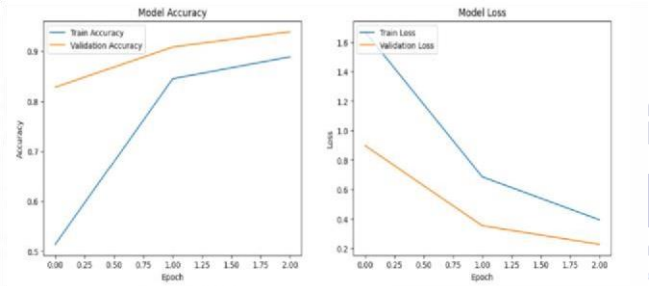

Figure 7 FNN Model Accuracy and Loss Graph

Table 1 Comparison of Accuracy of all Model

Model	Accuracy
RNN	95.61%
FNN	93.89%
CNN	84.86%

5.4. Explanation

RNN is much better because it have higher accuracy than a CNN and FNN. When we apply these three models on our dataset the most accurate result given by the RNN model

Model	Accuracy		Model	Accuracy
RNN	95.61%		Poly kernel	86.34
FNN	93.89%	VS	RBF kernel	86.32
CNN	84.86%		Normalized poly kernel	82.84

and it is use for the long short term memory loss as well

5.5. Reasons for RNN

RNN may be a more suitable choice based on the provided metrics. Here are some reasons to choose an RNN:

1) High Recall:

The RNN demonstrates a significantly higher recall compared to the CNN. If correctly capturing instances of the positive class is crucial for your application, a higher recall indicates that the RNN is better at identifying these instances.

2) Balanced F1 Score:

The F1 score, which is the harmonic mean of precision and recall, is higher for the RNN compared to the CNN and FNN. This indicates that the RNN provides a better trade-off between precision and recall, which makes it suitable for situations where false positives as well as false negatives should be reduced.

3) Accuracy:

The other model is less accurate than the first, which has an accuracy of 95.61 %. While higher accuracy seems to be an indicator of better overall performance, it's not the only metric that you should pay attention to.

4) Precision:

Precision is the ratio of accurately predicted positive observations to all predictions. The former is relatively precise, while the latter employs a model with lower precision. Precision counts when accuracy in detecting false positives matters.

5) Sequential Data Handling:

RNNs are designed to function with sequential data, and can therefore be used for actions where the order of input elements is an important consideration. But if there are temporal dependences or sequential patterns, an RNN is usually better at getting a handle on them.

6) Application to Time-Series or Sequential Data: Time-series data, natural language or any other sequence of things should be seen as being more suitable for the

RNN approach. Examples include tracking tasks like language modeling, sentiment analysis and speech recognition.

7) Potential for Long-Term Dependencies: However, RNNs do have the ability to learn end-to-end from sequential data with long dependencies. Vanishing gradients are a possible problem? Architectures such as LSTM (long short-term memory) or GRU (gated recurrent unit) can alleviate these difficulties.

8) Model Complexity:

Yet RNNs, which excel at processing sequential data, are known to be simpler and more costly. Trade off between complexity and performance must be assessed based on the size of the data set available for use, as well as resources.

9) Flexibility in Sequence Length:

Because RNNs are able to accept input sequences of variable length, this is helpful if your instances vary in length.

5.6. Comparison

Table 2 Comparison of Model that use earlier

5.7. Explanation

The RNN yielded the highest accuracy of all models at 95.61 %. RNNs are especially adept at capturing the sequential dependencies and patterns in data. A more straightforward feedforward neural network, the FNN also did well with an accuracy of 93.89 %.The CNN, for which the standard tasks include those involving grid-like data such as images was done with an accuracy of 84.86 %. In the second set of models using different kernels: The Poly Kernel had an accuracy of 86.34 %, only just a shade better than the RBF Kernel at 86.32 %. These are typical kernels used in SVMs as well as other kernelized models. The Normalized Poly Kernel reached an accuracy of 82.84 %.

Looking at this comparison, the RNN has a higher accuracy. The Recurrent Neural Network (RNN) is preferable because it can capture the sequential dependencies in data and so performs well on tasks where temporal relationships are important. Because the RNN has this innate capability of modeling sequential patterns, it attains the highest accuracy in comparison with all other models (95.61 %).

6. CONCLUSION

Thus, RNN offers a specific architecture for dealing with sequential data that grasps the direction or trend inside some span of time. It also has a shape that creates an internal feedback loop so the results from certain time periods can return to earlier ones. For this reason, they are versatile for a whole range of tasks such as natural language processing, time series and speech recognition.

Of course, RNNs have their drawbacks as well and suffer from a difficulty of the vanishing gradient which makes it difficult for them to learn long-term dependency. These issues have seen the development of more advanced variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

The new models improve the ability of RNNs to store significant data across long periods. Another notable feature of RNNs is their ability to handle sentences of varying length. Their application would be relevant for situations that involve order and context of information.

Although RNNs are a powerful way of addressing sequential data, one must be very thoughtful about the nature of the task and the characteristics of the data. Consideration needs to be made on what concerns computational workload related to sequential processing as well as influence the length of sequence on performance. Researchers are looking for ways to improve the deep learning field by building more versatile solutions which address more complex problems such as hybrids of RNNs and CNNs. However, the real value of RNNs rests in designing a suitable architecture, optimizing parameters, and gaining insight into the peculiarities of any particular case.

6.1. Future Work

However, there are still many challenges facing the refinement of the RNN architecture, namely working out vanishing gradients problem as well as the increased computational and time load. Some researchers are constantly working to make the deep learning field much better through the coming up of more flexible systems such as hybrids between RNNs and CNNs. The

second generation hybrids seek to address larger issues using an integration or combination of characteristics taken from architecture.

Therefore, the future research on RNNs should not concentrate on the refined model development only. The process should focus on an evaluation of the kind of activities in question, as well as the peculiarities inherent in a particular kind of data. The parameter optimization as well and appropriate architecture for each case will remain important. Lastly, understanding the relationship between computational load, sequential lengths, and performance is also crucial to the continued use of RNNs in many areas. However, it is not just about RNN per se, rather the clever use of those capabilities for addressing the practical issues.

REFERENCES

A. a. J. J. M. 2. I. o. T. i. d. s. A. c. r. a. f. d. C. C. p.-2. Heidari, "Internet of Things intrusion detection systems," Cluster Computing, 2022.

A. K. M. J. R. R. R. S. T. a. B. S. Khan, "Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions. Security and Communication Networks," 2022.

B. A. M. S. P. a. A. A. Dash, "Threats and Opportunities with AIbased Cyber Security Intrusion Detection:," A Review. International Journal of Software Engineering & Applications (IJSEA), p. 13(5), 2022.

X. L. T. L. D. W. J. W. Y. Z. H. H. Zhen Yang, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection,," Computers & Security, vol. Volume 116, 2022.

A. A. Shah, "A Machine-Learning-Based Approach for Autonomous IoT Security," A Machine-Learning-Based Approach for Autonomous IoT Security, p. 17, 2020.

M. F. K. Ahmed, "An intrusion detection system for packet and flow based networks using deep neural network approach," International Journal of Electrical and Computer Engineering, 2020.

C. J. D. W. L. W. W. L. F. a. Y. A. 2. C. r. o. n. i. d. m. b. o. m. l. C. &. S. p. Zhang, "Comparative research on network intrusion detection methods based on machine learning," Computers & Security, 2022.

"Analysis of Machine Learning Techniques for Intrusion Detection System," Analysis of Machine Learning Techniques for Intrusion Detection System, p. 10, 2019.

E. Z. A. U. M. a. A. A. Mushtaq, "A two-stage intrusion detection system with auto-encoder and LSTMs.," Applied Soft Computing, pp. 121, p.108768., 2022.

"UNSW_2018_IoT_Botnet_Final_10_best_Testing, " intrusion detection in iot, p. 26.

Y. D. Y. C. Z. L. Q. a. X. W. Fu, "A deep learning model for network intrusion detection with imbalanced data," Electronics, p. 898, 2022.

M. A. O. A. A. A.-O. S. B.-H. N. H. A. A.-Z. A. L. A. A. a. A. T. Almaiah, "Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels.," Electronics, 11(21, p. 3571, 2022.

A. A. S. L. U. S. S. M. P. A. A. E. H. a. R. K. Balyan, "A hybrid intrusion detection model using ega-pso and improved random forest method," Sensors, 2022.

E. A. M. V. C. a. P. G. GSR, "Hybrid optimization enabled deep learning technique for multi-level intrusion detection," GSR, E.S., Azees, M., Vinodkumar, C.R. and Parthasarathy, G., 2022. Hybrid optimization enabled deep learning technique for multi-level intrusion detection. Advances in Engineering Software, 173, p.103197., 2022.

Y. A. A. M. S. H. M. a. C.-P. R. Saheed, "A machine learning-based intrusion detection for detecting internet of things network attacks.," Alexandria Engineering Journal, pp. 9395-9409, 2022.

M. H. K. I. M. U. M. A. A. Y. M. A. F. a. M. M. Talukder, "A dependable hybrid machine learning model for network intrusion detection," Journal of Information Security and Applications, pp. 72, p.103405., 2023.