JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Deep Learning Model for Intrusion Detection in Wireless Sensor Networks

Pooja Raghav [1], Kamal Kumar [1], Gautam Kumar [2]

[1]Pooja Raghav, M. Tech Scholar, Department of Computer Engineering, SDDIET, Kurukshetra University, Haryana, India
[1]Kamal Kumar, Assistant Professor, Department of Computer Engineering, SDDIET, KUK University, Haryana, India
[2] Gautam Kumar, Assistant Professor, Department of Computer Engineering, SDDIE, KUK University, Haryana, India

***Abstract*:** Wireless sensor networks have been given several security procedures or tools (WSNs). To ensure defense in depth, an intrusion detection system (IDS) should always be installed on security-critical applications. The intrusion detection method for conventional networks cannot be utilized directly in WSNs due to resource limitations. For the purpose of finding intrusions in wireless sensor networks, many approaches have been put forth. However, the majority of them concentrate on targeted attacks (such as selective forwarding) or attacks on specific layers, like the media access layer or the routing layer. In this study, we provide a framework for wireless sensor networks' intrusion detection systems that is based on machine learning. Our system will not be restricted to specific assaults, and machine learning algorithms assist to automatically develop detection models from training data, saving human work from generating attack signatures or defining the typical behavior of a sensor node.

Keywords: Intrusion detection, SVM, Deep learning, WSN

I. INTRODUCTION

IoT tactics for collecting sensor data and creating intelligent applications and services have attracted a lot of interest in recent years. The Internet of Things (IoT) is described as the process of connecting any thing to the internet using integrated software and hardware that allows for data collection, exchange, and communication. The deployment of the Internet of Things makes the world much more accessible and offers an almost infinite variety of possibilities and interactions at home, at work, and during leisure time. With the help of the Internet of Things, people, sensors, and gadgets are connected, creating a form of fluid interaction between hardware and software.. In the same way that networks and computer monitors are used on the internet to enhance the element of organizations owing to the development of Machine learning and artificial intelligence, these interactions allow gadgets to anticipate, react, respond, and improve the physical world. By connecting sensors, devices, and people, the Internet of Things creates a kind of fluid interaction between hardware and software as well as between humans and machines. Similar to how networks and computer monitors are utilized on the internet to enhance the element of organizations owing to development of Machine learning and artificial intelligence[1], these interactions allow gadgets to anticipate, react, respond, and improve the real environment.

implement network connectivity, sensors, actuators, necessary electronics, software, and other components to enable communication between all linked things. The IoT network assigns a unique identity and IP address to each object. It's referred to as M2M [3]. The items or related equipment to this network communicate information, allowing numerous tasks to be completed according to the system owner's specifications.

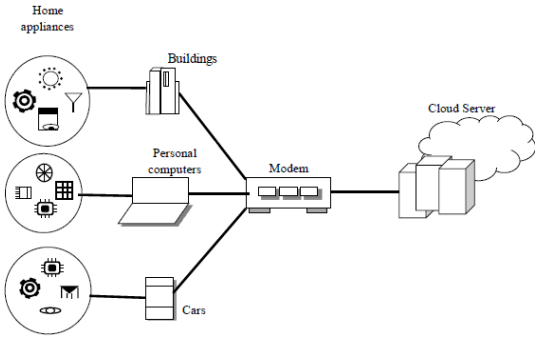


Figure 1: IoT Paradigm Represented Diagrammatically

A. Security in IoT

. In the realm of information technology, security concerns have long been prominent. Regularly, new and distinct security concerns arise as a result of characteristics of IoT deployments with many functionalities. In order to ensure the security of IoT products and services, it is crucial to first and primarily resolve these problems. The devices and data services that are connected to the Internet of Things must be secure for their users in order for it to function. In a time where technology is omnipresent and

present in practically everyone's everyday activities, this is crucial. Iiot equipment and goods might be used by hackers as a point of entry. Users' private information is at danger, and hackers can take advantage of this by lowering the security of data streams. Due to the interconnectedness of Internet-connected devices, any one with a low degree of security has the potential to jeopardize the dependability and security of the Internet as a whole. The spread of homogeneous IoT devices, certain devices' ability to connect to other devices automatically, and the possibility for launching these devices into unsafe locations all contribute to the complexity of the situation[5]. IoT equipment and software users and developers have a responsibility to work together to protect consumers and the Internet from any potential harm. " In order to provide effective and appropriate solutions to Internet-of-Things security concerns that are well-suited to their size and complexity, a collaborative security paradigm will be required.

1. Real-Time Intrusion Detection System

It is presently in development and makes use of deep learning, neural networks, data collection, and other methodologies. For the use and research of intrusion detection systems, it has distributed a number of clever methodologies. The study's main goals are to decrease the possibility of near-misses, identify false alarms, and improve the system's capacity for self-learning.. Research on intrusion prevention is increasingly concentrating on distributed, highly accurate, quick, and intelligent detection. and will include the methods detailed below.

- *Distributed Intrusion Detection*- Distributed systems, which utilize collaborative processing, dispersed structure, data analysis of a range of data types, as well as a single intrusion detection system design compared to better detection capabilities, are frequently used in heterogeneous networks and huge networks.
- *Intelligent Intrusion Detection*-. It is presently in development and makes use of deep learning, neural networks, data collection, and other methodologies. For the use and research of intrusion detection systems, it has distributed a number of clever methodologies.
- *High-Speed Packet Capture Technology*- Network intrusion detection systems can benefit from high-speed packet capture to accelerate detection and use less resources..
- *Efficient Pattern Matching Algorithm*- The rules must be able to accommodate complicated models as invasions get more diverse and sophisticated. Therefore, it is crucial to enhance and improve the pattern matching engine..

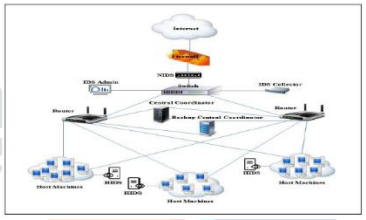


Figure 2 A collaborative (IDS) framework for a big data environment

A subfield of artificial intelligence (AI) known as machine learning (ML) research how computers might learn to modify their behavior based on earlier observations. Researchers from a wide range of fields, including computer science, engineering, mathematics, and cognitive science, have been interested in these tactics because they allow individuals to adapt to their environment and learn from their past experiences. In general, a learning strategy transforms information learned from the outside world into a new form that may be used later.

The two most common categories of machine learning algorithms are classification and prediction. In this part, the specifics of analysis and regression are briefly discussed.

2. Classification

- Learning and classification are two-step processes, according to data mining literature.
- Learning: A classification model analyses data for training, and classification rules are used to specify the learned model or classifier.
- Classification: The accuracy of the classification rules is evaluated using data. If the correctness is judged satisfactory, the rules applied to the classification of new data tuples. Because each data tuple contains a class label, this process is also known as supervised learning. Support vector computers, artificial neural, decision forests, KNN, and genetic programming are examples of supervised learning methods (Gas)

A classifier's confidence the classifier's accuracy in classifying the tuples in a given test set is stated as a percentage, in contrast to unsupervised (clustering), where each training tuple's class label is unclear and the number or set of categories to be taught is indeterminate.

3. Supervised Learning

- Decision trees - A decision tree helps a person navigate a list of possibilities based on their current choice. The usage of trees to represent this kind of decision-making process. From the tree's base to each individual specimen's last leaf node, specimens are categorized. Each node is given a specimen-specific trait, and the number of bank branches assigned correlates to the result. Just a few examples are CART (Classification and Regression Tree), C4.5, and ID3.
- K-Nearest Neighbor Analysis (K-Nearest Neighbors) The most basic and popular nonparametric method for identifying specimens is nonparametric identification (k-NN). It simply requires input vectors to calculate the separations between different places, after which it categorizes the unknown location into the K-nearest immediate neighbors class.. The training data stage is therefore skipped in favor of finding and categorizing input vector occurrences. k-NN "trains" and "identifies" occurrences just as a consequence "online"

B. Unsupervised Learning (Cluster Analysis)

- . It groups objects into conceptual and physical groups that are connected. Data from one cluster are comparable to other clusters' data but not the other way around.
- . A Density-Based Methodology - According to their density, these items are grouped together in this method. It either divides items into groups based on their proximity (as in DBSCAN) or on some posterior distribution (e.g. DENCLUE). Beam is a density-based method for enhancing the ordering of data clustering structures.

The approach begins by grouping the collection of data pieces that need to be processed. This approach divides the object space into a finite number of cells, and then clusters the regular grid (e.g. STING is a typical example on statistical information stored in grid cells). CLIQUE and Shockwave Cluster are two strategies for concentrating and collecting energy.
This approach automatically creates a model for each group, and then finds the data that best matches the model (e.g. EM algorithm).

1. Prediction

Forecasting continuous valued functions uses prediction models. We might create a categorization mechanism to classify bank loan applications as safe or risky, or a forecast tool to determine how much electronics potential consumers will spend based on their salary or line of work..

II. OBJECTIVES

This study has several goals, including: 1. To learn, evaluate, and use different safe routing algorithms used for detecting rogue devices by launching a version number attack in Analyze how an assault might affect the operations of the IoT network.
2. To develop a method for calculating energy level for trust-based processes.
3. To provide an algorithm for multipath routing approaches that can detect malicious nodes.
to put the recommended technique for identifying the attacker's firmware version in an IoT infrastructure into use and evaluate it against other ways on a range of criteria.

III. LITERATURE REVIEW

Ahmet Arış, et.al (2020) centered on comprehending the effects of many VNA in RPL-based IoT networks [12]. The impact of many attackers was examined from a variety of perspectives. Based on simulations and analysis, it was shown that only the PDR was impacted by the maximum number of attackers, whereas neither the average network latency nor the average power consumption were. The results showed that greater delays were experienced when attacking locations were closer to the root, and better PDR results were produced during center assaulting positions by using more power. Finally, a mitigation method that was recently developed has its effectiveness tested against a range of potential attackers..
Ş. Okul, et.al (2017) . theoretically explained the importance of the IoT idea. Although the Internet of Things did not have a comprehensive layer structure, it was determined that it had three common levels: the object, network, and application layers [13]. Additionally, security epidemics like Botnet, Social Engineering, DID, MIM, and DOS were explored with examples and analyses for the Internet of Things. Finally, these assaults also outlined the safety measures that were crucial for Internet of Things layers..

IV. METHODOLOGY

A. Convolutional Neural Network (CNN)

In fact, CNN is a commonly used DL paradigm based on the visual brain of an animal. Convents are a kind of acyclic neuronal array-like neural network. A NN varies from a NN in that a neuron in the secret level is only connected to a portion of the neurons in the layer above it. Due to the sparse connection, it is able to learn implicit attributes. In a hierarchy extract function, qualified first-level filters were identified as a series of edges and color blobs, qualified second-level filters as figures, qualified third-level filters may identify item components, and qualified fourth-level filters can categorize those objects, as shown in Figure 3..

1. Convolutional Layer- .

This layer contains computations as its primary unit. It is made up of many neuron identity maps. A layer parameter is a collection of kernels or filters that may be studied. They work in conjunction with a two-dimensional activating map on top of which is placed an output volume activating map. The weight of neurons on the same map reduces the dimensionality of a network (constantly shared). The sparse associations between two-layer neurons are extended by perceptrons, which are hyperparameters. The stride (the direction of the filter's movement) and depth (the number of layer filters) are two of the three hyperparameters that regulate volume size (to monitor 54 output space). The ConvNets are capable of propagating in reverse.

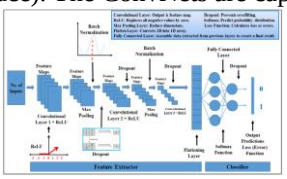


Figure 4. - Basic Conv Net Architecture

2. RELU -

ReLu is a quasi-function that replaces all negative image pixels in the feature map with a value of zero. Because the rest of the knowledge we need to know from the actual world will not be linear, ReLU in the CNN has the goal of not being linear. Repaired piecewise linear functions, or those that threshold at 0, are distinguished from:

3. Pooling Layer (PL)-

These functions restrict the number of parameters inside a network and the determination of activation maps (without compromising data), decreasing overall device complexities. The basic ConvNet structure consists of alternating conv or group layers. This is an examination of the overfitting issue. Some of the bundling techniques include max, average, stochastic pooling, and multi-scale order less pooling. The spatial width F(l) and stride S of the filter are the two hyperparameters that make up

Layer 1. (l). Equation states that it produces a result of size m 1((l))m 2((1))m 3((1)) from an input of size m 1((l))m 2((1))m 3((1)) (2).

where;

$$m_1^{(l)} = m_1^{(l-1)}$$

$$m_2^{(1)} = m_2^{(l-1)} - F^{(1)}/S^{(1)} + 1$$

$$m_3^{(1)} = m_3^{(l-1)} - F^{(1)}/S^{(1)} + 1 \quad (2)$$

4. Max pooling:.

It is sometimes referred to as space pooling, down sampling, or subsampling. Each feature map is condensed while keeping the most crucial details. Average, mean, and number are all instances of maximum pooling. A spatial neighbourhood is the most crucial component of the corrected function modelling of that window in the case of Max Pooling (for example, a 2 x 2 window). We might aggregate all of the window's components using the conventional technique (mean pooling), as opposed to making the biggest component. Sometimes, it seems that pooling as many resources as you can will be the most effective use of those resources.

5. Fully Connected Layer –

FCL is used for classification jobs. For categorisation, it employs a softmax activation function. According to the term FC, every neuron in the preceding layer is connected to every neuron in the next layer. Higher-level characteristics are denoted using the outcome from earlier levels. This layer's main objective is to categorize the input picture using higher-level data.

f(x)=max (0, x)

6. Loss Layer-

A losses layer known as FCL determines loss or error and penalizes deviations from the desired result. A single equally exclusive class from K is calculated using the SoftMax cost formula. It's a typical sign of failure. It is a logistic equation with several nodes. By converting projections to non-negative values, it is possible to achieve probability distribution over groups more generally. Hinged failure is calculated using Vector Machine Support, a broad margin classifier. Euclidean failure should be applied in order to go back to real-value labels[4].

The neural network (NN) is composed of a series of connections that represent the activity of the brain. Each node has a weighted link with a variety of other nodes in adjacent levels...

Learned convolutional kernels or filters are used by CNN as a feature extraction technique. The major goal is to use learnt features as the feature representation for a recommendation rather than pre-designed qualities. Deep CNNs are newly built convolutional networks with all of their layers connected. Higher-level features are frequently learnt as the network depth increases, and discriminative elements become more salient.

Finally, we provide a technique for recommending movies that is based on the results of several research studies.. The system will make recommendations for films that are comparable to the one the user just completed viewing when they have finished watching the necessary video.

B. Proposed Algorithm-

1. Begin
2. Data gathering form CIC CSE KDD99 data from http://205.174.165.80/CICDataset/NSL-KDD/ with 494020 rows × 124 columns.
3. Extract data using target type like dos, r2l, u2r, probe and normal.
4. Map actual type to another column called 'target_type' and check missing values.
5. Explore categorical feature using get numeric data.
6. After process data apply EDA and perform standart deviation over dataset.
7. Using graphs and charts to better comprehend the data, like Services, target type, flag on kdd cup, protocol type.
8. Use the methods get_dummeis to conduct data pre-processing to extract the required features to convert categorical to numerical values. Apply one hot encoding and label encoding on features column.
9. Split data into train test using train_test_split into 80:20 ratio.
10. Create a neural network, train it on data, and define it using the keras and tensorflow libraries.
11. Assess the model and generate matrices to illustrate the accuracy and loss graphs.
12. Stop

C. Proposed Methodology-

The CIC KDD NSl dataset (https://www.unb.ca/cic/datasets/nsl.html), which offers two files for training and assessing machine learning approaches, was used to train and test the proposed model. These files are KDDTrain.txt and KDDTest.txt. Finally, we created a panda data frame from the.txt data that has 42 textual and numerical properties. Following that, descriptive statistics in terms of count, mean, standard deviation, minima, and maxima were computed and presented for a Python frame using the describe() function, which supplied statistical information. Additionally supported are data frame columns and Pandas series objects.. Count the number of distinct assaults in the data frame. Using univariate, bivariate, and multivariate features for visualization, data for analytical attacks (normal, dos, probe, r2l, and u2r) on different networking payload protocols, such as TCP, UDP, and ICMP, were analyzed. To assess density, we used a bar graph analysis of the label, protocol, flag, and duration graphs.

4.5 Perform EDA

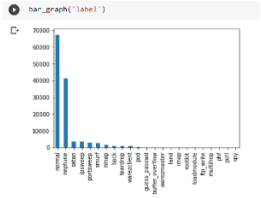


Figure 5: Bar Plot of Sub Attacks.

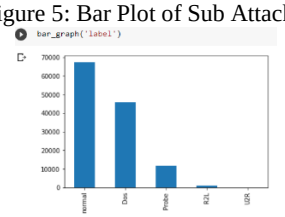


Figure 6. Bar Plot of 5 Main Attacks.

. Figure 5 displays the multivariate bar graph evaluation for the label. The value counts for each assault, such as normal, nNeptune devil, ipsweep port sweep, and so on, are shown together with the number of attack labels. Figure 6 s bar graph analysis for labels shows the number of counts for common, Dos, Probe, R2L, and U2R attack types.

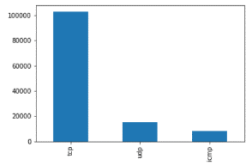


Figure 7: Plot of Protocols Used.

A study of many numbers for several network payload protocols, including TCP, UDP, and ICMP, is shown in Figure 7. After data for various frequencies were binned, the frequency duration histogram is displayed in Figure 8. The image showed a bar graph showing the number of different flags. There were counts for the following flags: SF, S0, REJ, RSTR, RSTO, and others. Bivariate analysis was used to assess the samples in a dataset, as well as the attack type and strategies used. The quantity of samples in a dataset is also important, as are the services and attack type. This is what the 1323 matrix produced..

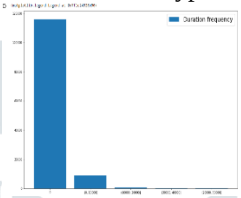


Figure 8: Frequency of Duration.

To prepare data for training and testing for binary and multi-class classification, we employed Label and One-Hot encoding. Sk-learn, a potent method for converting category feature levels into numbers, has been applied in a number of ways. A software called Label-Encoder can encode labels with values between 0 to n classes-1, where n is the total number of labels. The representation of categorical variables as binary vectors is one example of an encoding. The categorical values first need to be converted to integers. Then, a binary vector with all zero values is created for each integer value, with the exception of the integer's index, which is marked with a 1.

feature choice,. The final forms of X and Y were, respectively, (125973, 42) and (125973, 42). (125973, 1). A character's significance acts as a connection. The percentage of samples in the dataset that used the TCP protocol type and caused an incursion as well as the percentage of samples that used the TCP protocol type and caused a normal condition.

D. Proposed Flowchart

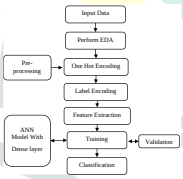


Figure 9 – Proposed Flowchart

A CNN-based RS, as seen in Figure 9, consists mostly of the following procedure, which makes recommendations for the target user. The input layer receives the data and preprocesses its attributes after that. The pre-processed data is then used to produce each typical feature vector and feature extraction from the embedded layer. The full-connection layer runs the software to connect attribute features and produce user and item attributes after the embedding process. The predictions attack is then created using the independent and dependent properties. The model for binary classification is trained using the Deep ANN Classifier. 29 neurons are used to train the sequential model in the Dense layer, and the Softmax activation function in the output layer completes the model.

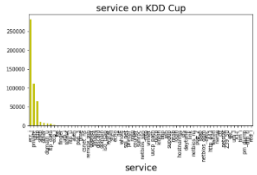


Figure 10 – Services used in KDD Cup dataset.

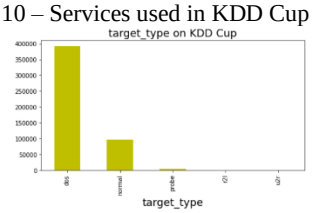


Figure 11 – Shows target type (5) or attacks.

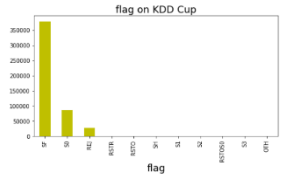


Figure 12 – Shows Flags used in attacks.

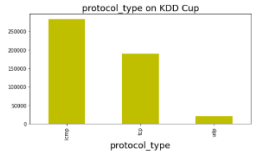


Figure 13 – Shows Protocol used in dataset.

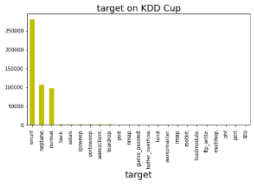


Figure 14 – Shows 13 target type or 13 attacks.

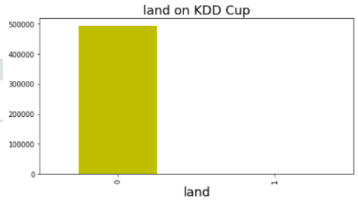


Figure 15 – Shows Land on KDD cup.

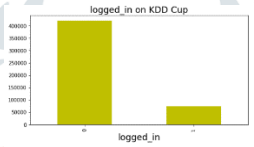


Figure 16 – Shows Logged in on KDD cup.

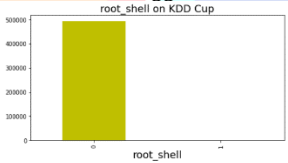


Figure 17 – Shows Root Shell on KDD cup.

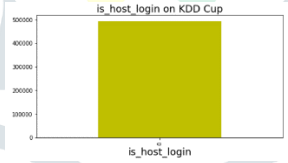


Figure 18 – Shows Host Login on KDD cup

Figure 19 – Shows Guest Login on KDD cup

Figure 20 – Shows Standard Deviation on Numeric Features .

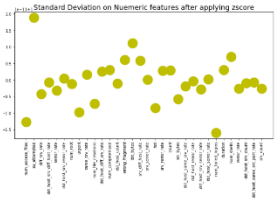


Figure 21 – Shows Standard Deviation on Numeric Features after Zscore.

V. SYSTEM IMPLEMENTATION

A. PYTHON

A lot of people utilize the programming language Python. Guido van Rossum created and released it in 1991. Python produces high-quality artifacts, understanding, and interactivity. Python is meant to be an easy language to learn. While others utilize punctuation, this makes significant use of English terms. It features fewer syntactic structures than other languages..

• Python is interpreted: The interpreter runs Python at runtime. Before you put your program into use, you don't need to enhance it. PHP and PERL have a lot in common.

Python is a programming language for absolute beginners: the program Beginner programmers have access to a wide range of queries in Python for simple WWW application word processing for sports..

In contrast to other programming languages like C, Fortran, or Java, Python is a language that lets the user to focus more rapidly on solving domain issues rather than stumbling over the intricacies of how a machine operates. Python succeeds in achieving this goal thanks to the following qualities:

Python is a high level language, which implies it summarizes the technical information about computers.

• Python has the main library but many third-party implementations, which provide a wide range of popular codebases and models of problem-solving.

• The programmers can quickly find solutions and sample code for problems with the help of Google and Stackoverflow.

• Python has several users.

To demonstrate:

```
Garage = "Ferrari", "Honda", "Porsche", "Toyota"

 print(each car) for each car in Garage
```

"print()" is an optimized Python function that produces console text.

• Building predictive models with enough data helps increase model accuracy.

It may seem obvious that a "Ferrari, Honda, Porsche, Toyota" console may expose something as "printing" transmits text to the "console."

What tasks can Python perform? Python is a powerful programming language that can do practically any other language and has a similar speed.

Python will thread or process GPUs just like any other language. Additionally, most C/C++ code is wrapped in Python by data processing modules.

Python is a fantastic choice for mathematical computations because we can create code simply, check it rapidly, and because its syntax reflects how mathematical ideas are described in the literature. By learning Python, you will have access to a vital tool for many software engineers. Python implementations in the real world:

- GUI-Based Desktop Applications
- Web Frameworks & Web Application
- Enterprise & Business Applications
- Operating Systems
- Language Development
- Prototyping

1. Python Environment Variables

- *Python path:* . It plays a role akin to PATH. This variable instructs the Python interpreter how to locate files from imported modules. Either the Python source code or source library folders will be present. Python will occasionally predetermine PYTHONPATH.
- *Pythoncaseok:* An import declaration in Windows contains the first case-insensitive match. To make this variable allowable, provide a value.
- *Pythonhome:* This quest line leads to a different module. It is included in the PYTHONSTARTUP or PYTHONPATH directories to simplify changing out module libraries.

B. Jupyter

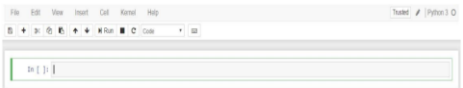
A popular tool for interactively generating or viewing data science projects is the Jupyter Notebook. A notebook combines code or its output with a picture, text, mathematical formula, or other rich media in a single document. Intuitive workflow promotes quick, iterative development, increasing the likelihood that notebooks will be a contender for the core of contemporary data science, technology, and innovation.

pip only can be used when you are an experienced Python user or prefer manual package management.

```
pip3 install jupyter
```

1. The Notebook Interface

Hoping the interface doesn’t appear foreign given the fact that in front is an open book. Jupyter is actually a competent word processorIn a short while, the menus (or Ctrl + Shift + P) may be used to display the list of instructions that are available, and which are the little buttons with either symbol.



2. Cells

We'll return to kernels a little later, but first, let's get cells under control. Cells make up a notebook's main body. In the screenshot of a notebook in the previous section, the box with a green border represents an empty cell. We address the following two main categories of cells:

• A code cell contains instructions for the kernel to execute, or its result is shown in the next section.
• A Markdown cell generates Markdown-formatted text and immediately displays its result.
A new notebook's first cell is always a code cell. Let's give it a try using the well-known example of a hello world. In a cell in the toolbar above, tap the run button, or press Ctrl + Enter. This ought to be an

```
print('Hello World!')
Hello World!
```

The output of a cell is displayed below, and the label to the left has changed from In[] to In[1]. You can see the output of the code cell in this item since it is incorporated in the text. Because coding cells have that label on the right and markdown cells do not, it is always feasible to distinguish between the two types of cells. Creating your first new code cell

```
importtime
time.sleep(3)
```

This cell runs for 3 seconds without producing any output. Keep in mind that Jupyter indicates that a cell is running when its label is changed to [*]. The value of the final line in a cell, whether it be a single attribute, functions call, or something else, determines the output of a cell type as well as all text information that is expressly written during cell execution. For example, [7].

```
defsay_hello(recipient):
return'Hello, {}!'.format(recipient)
say_hello('Tim')
'Hello, Tim!'
```

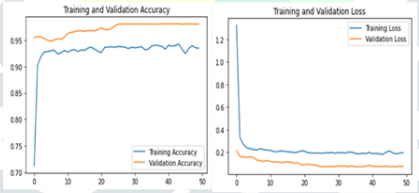
VI. SIMULATION AND RESULTS

A. Modelling of Neural Network

Table 1: Parameters used for Training

Model	Sequential
RNN	ANN
Training data	80%
Testing data	20%
Data shape	(81581,42)
Layers	DENSE with 29 neurons
Output layer	Dense with 5 neuron
Output function	SOFTMAX
Loss function	Sparse categorical cross-entropy
Optimizer	ADAM
Metrics	ACCURACY

Following model training, Figure 22 shows the accuracy vs epoch graph for the train and test datasets, along with the graph plot of loss against epoch for the train and test datasets.



Accuracy and Loss Graph of Train and Test Data, Figure 22
The recommended ANN model outperforms the other ML approaches, as shown in table 2, which shows this. The ANN model's accuracy is 94.45%; its precision is 96.38; its recall is 98.94; and its F1 score is 97.64. Results from different machine learning models, including Decision Tree, Support Vector Machine, and others, were contrasted with those from our ANN model.
One of the performance metrics used to evaluate the result is the F-measure, along with the accuracy, precision, recall, and F-measure.
A. Accuracy -. It is a fundamental accuracy metric that is related to the overall number of measurements. Statistical accuracy is enhanced with symmetrical datasets due to the roughly equal proportion of false negatives and false positives.
Accuracy = TP+TN/TP+FP+FN+TN
B Precision - It measures how many positive observations were successfully anticipated out of all the positive observations expected.
Precision = TP/TP+FP
C. Recall (Sensitivity)- Yes, it's a ratio of favorable comments that was precisely anticipated based on all of the actual class observations.
Recall = TP/TP+FN
D F1 score - An average of recall and accuracy that is weighted. Therefore, this score takes into consideration both false positives and false negatives.
Table 2: Model Matrices and Performance Evaluation.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Propose DeepANN	**93.46**	**98**	**91**	**96**
DT	88	85	94	89
SVM	85	85	87	86

VII. CONCLUSION

The identification of specific nodes is a significant difficulty for WSN practical applications. WSN is especially susceptible to network assaults because to its growing service area and volume of data. Many of today's ID systems are worthless in the face of novel and unforeseen dangers since they can only defend against particular sorts of assaults. It is challenging to give WSN security organizations based on IDS accurate threat detection tools. We describe an intelligent ID technique that makes use of progressive machine learning. With the use of a cluster WSN network topology, the model finds intrusions and then categorizes them according to their nature. On the KDD99 dataset, we trained and tested our suggested model in this research...

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A Survey," 2010, Computer Networks, vol. 54, no. 15, pp. 2787–2805

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," Elsevier Future Generation Computer System,Vol. 29, No. 7, pp. 1645–1660, 2013

[3] Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," 2015, Ad Hoc Networks, vol. 28, pp. 68–90O. Said and M. Masud, "Towards internet of things: survey and future vision," 2013, International Journal of Computer Networks, vol. 5, no. 1, pp. 1–17

[4] J. S. Kumar and D. R. Patel, "A survey on internet of things: Security and privacy issues," 2014, International Journal of Computer Applications, vol. 90, no. 11.

[5] R. M. Cardoso, N. Mastelari, and M. F. Bassora, "Internet of Things Architecture in the Context of Intelligent Transportation System – A Case Study Towards a Webbased Application Deployment," in 22nd International Congress of Mechanical Engineering (COBEM 2013), 2013, pp. 7751–7760

[6] T. Abels, R. Khanna, K. Midkiff, "Future Proof IoT: Composable Semantics, Security, QoS and Reliability", 2017, in Proc. of IEEE Topical Conference on Wireless Sensor & Sensor Networks (WiSNet), pp. 1-4

[7] Chandni, Rakesh Kumar, "Trust Based Technique for the Mitigation of Version Number Attack in Internet of Things", 2019, International Journal of Recent Technology and Engineering (IJRTE), Volume8 Issue3

[8] M.V.R Jyothisree, S. Sreekanth, "Attacks in RPL and Detection Technique used for Internet of Things", 2019, International Journal of Recent Technology and Engineering (IJRTE),Volume8, Issue

[9] Abhishek Verma and Virender Ranga, "Analysis of Routing Attacks on RPL based 6LoWPAN Networks", 2018, International Journal of Grid and Soft Computing, Volume 11, Issue 8, pp. 43-56

[10] Ahmet Aris, Sema F. Oktug, S. Berna Ors Yalcin, "RPL version number attacks: In-depth study", NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium

[11] Ahmet Arış, Sema F. Oktuğ, "Analysis of the RPL Version Number Attack with Multiple Attackers", 2020, International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)

[12] Ş. Okul, M. Ali Aydın, "Security Attacks on IoT", 2017, International Conference on Computer Science and Engineering (UBMK)

[13] Ruchi Vishwakarma, Ankit Kumar Jain, "A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks", 2019, 3rd International Conference on Trends in Electronics and Informatics (ICOEI)