**A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks**

**ABSTRACT:**

Deep learning approach plays more important role in the data science, when it comes to the large datasets. In this paper we are analysing Intrusion detection attacks. Intrusion detection plays an important role in ensuring information security, and the key technology is to accurately identify various attacks in the network. In this paper, we investigate how to display an interruption location framework dependent on profound learning, and we propose a profound learning approach for interruption discovery utilizing recurrent neural networks (RNN-IDS). In this project we are going to analyse the KDD datasets which consists of 44 feature based on feature we are going to apply the classification algorithm(Recurrent neural network) which helps in training the data and helps in finding the accuracy. We compare it with those of decision tree, support vector machine, and other machine learning methods proposed by previous researchers on the benchmark data set

**Introduction:**

Nowadays Machin Learning is play more important role in the business as well as in scientific. Machine learning comes with many technologies like deep learning, which helps classification techniques.it helps in recommendation process easily. IN  recent years, network attack detection attracts increasing interest in social networking information security as the increasing security threats. With the inexorably profound reconciliation of the Internet and society, the Internet is changing the manner by which individuals live, study and work, yet the different security dangers that we face are turning out to be increasingly genuine. An Intrusion Detection System (IDS), a huge research accomplishment in the data security field, can distinguish an attack, which could be a continuous intrusion or an interruption that has just happened. In this paper we are distinguishing whether system traffic conduct is ordinary or abnormal, or a five-classification arrangement issue, i.e., recognizing whether it is typical or any of the other four assault types: Denial of Service (DOS), User to Root (U2R), Probe (Probing) and Root to Local (R2L). To put it plainly, the primary inspiration of interruption recognition is to improve the exactness of classifiers in adequately distinguishing the meddling conduct.

**Methodologies**

Recurrent neural networks include input units, output units and hidden units, and the hidden unit completes the most important work. The RNN model essentially has a one-way flow of information from the input units to the hidden units, and the synthesis of the one-way information flow from the previous temporal concealment unit to the current timing hiding unit is shown in Fig. 1. We can regard hidden units as the storage of the whole network, which remember the end-to-end information. When we unfold the RNN, we can find that it embodies the deep learning. A RNNs approach can be used for supervised classification learning.

Recurrent neural networks have introduced a directional loop that can memorize the previous information and apply it to the current output, which is the essential difference from traditional Feed-forward Neural Networks (FNNs). The preceding output is also related to the current output of a sequence, and the nodes between the hidden layers are no longer connectionless; instead, they have connections. Not only the output of the input layer but also the output of the last hidden layer acts on the input of the hidden layer.

**Motivation**

Developing absolutely secure systems is not possible

* Most existing systems have security flaws
* Abuses by privileged insiders are possible
* Not all kinds of intrusions are known
* Quick detection of intrusions can help to identify intruders and limit damage
* IDS serves as a deterrent

**Objective**

State the function of an Intrusion Detection System (IDS)

* Evaluate exterior and interior sensor placement effectiveness
* Consider trade-offs that influence exterior IDS design effectiveness
* Identify factors that influence interior sensor effectiveness

**scope**

Depending on their physical location in the infrastructure, and the scope of protection required, the IDS’ and IPS’ fall into two basic types: network-based and host-based. Both have the same function and the specific type deployed depends on strategic considerations.

**PROPOSED SYSTEM:**

**ADVANTAGES:**

* The RNN-IDS model not only has a strong modelling ability for intrusion detection, but also has high accuracy in both binary and multiclass classification.
* The model can effectively improve both the accuracy of intrusion detection and the ability to recognize the intrusion type.
* Helps in identifying the network traffic behaviour is normal or anomalous, or a five-category classification problem.
* Finding the network attack types.

**Existing System:**

* There exist either of one model in intrusion detection either binary or multiclass classification.
* Less Accuracy
* Difficult in understanding network Attack type

**LITERATURE SURVEY**

INTRUSION DETECTION SYSTEM – A STUDY Dr. S.Vijayarani1 and Ms. Maria Sylviaa.S 1Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore. 2M.Phil Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore. Intrusion Detection System (IDS) is meant to be a software application which monitors the network or system activities and finds if any malicious operations occur. Tremendous growth and usage of internet raises concerns about how to protect and communicate the digital information in a safe manner. Nowadays, hackers use different types of attacks for getting the valuable information. Many intrusion detection techniques, methods and algorithms help to detect these attacks. This main objective of this paper is to provide a complete study about the definition of intrusion detection, history, life cycle, types of intrusion detection methods, types of attacks, different tools and techniques, research needs, challenges and applications.

In this examination, an artiﬁcial insight (AI) interruption recognition framework utilizing a profound neural system (DNN) was explored and tried with the KDD Cup 99 dataset in light of consistently advancing system assaults. To start with, the information were preprocessed through information change and standardization for contribution to the DNN model. The DNN calculation was applied to the information reﬁned through preprocessing to make a learning model, and the whole KDD Cup 99 dataset was utilized to confirm it. At last, the precision, discovery rate, and bogus alert rate were determined to find out the location efﬁcacy of the DNN model, which was found to produce great outcomes for interruption recognition.[3][2]In this paper present an examination, routed to security pros, of AI strategies applied to the discovery of interruption, malware, and spam. The objective is twofold: to evaluate the present development of these arrangements and to recognize their primary restrictions that counteract a prompt selection of AI digital discovery plans. Our decisions depend on a broad survey of the writing just as on analyses performed on genuine undertaking frameworks and system traffic.[4] we propose a constant aggregate oddity identification model dependent on neural system learning. Regularly a Long Short-Term Memory Recurrent Neural Network (LSTM RNN) is prepared distinctly on typical information and it is equipped for anticipating a few time ventures in front of an information. In our methodology, a LSTM RNN is prepared with typical time arrangement information before playing out a live forecast for each time step.[8][10] To achieve high detection rate, data preprocessing, feature abstraction and multi-channel training and detection are seamlessly integrated into an end-to-end detection framework. Data preprocessing provides high-quality data for subsequent processing, then different types of features are extracted from the processed data.[6]

The simulation results show that our simulation system has a good approximation and can be used for intrusion detection in Tor networks.[1] In this paper, model of an interruption discovery framework is investigated dependent on profound learning, and Long Short Term Memory (LSTM) design is applied to a Recurrent Neural Network (RNN) and train the IDS model utilizing KDD Cup 1999 dataset. Through the exhibition test, it is affirmed that the profound neural system is successful for NIDS.[4] THIS paper shows the aftereffects of a writing overview of AI (ML) and information mining (DM) techniques for digital security applications. The ML/DM strategies are portrayed, just as a few utilizations of every strategy to digital interruption location issues.

The unpredictability of various ML/DM calculations is talked about, and the paper gives a lot of examination criteria for ML/DM strategies and a lot of proposals on the best techniques to utilize contingent upon the qualities of the digital issue to unravel.[6] In this paper, we present the application of Deep Recurrent Neural Networks (DRNNs) for prediction of user behavior in Tor networks. We constructed a Tor server and a Deep Web browser (Tor client) in our laboratory. Then, the client sends the data browsing to the Tor server using the Tor network. We used Wireshark Network Analyzer to get the data and then used the DRNNs to make the prediction. The simulation results show that our simulation system has a good prediction of user behavior in Tor networks.

an implementation of intrusion detection system using genetic algorithm Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas, Shahjalal University of Science and Technology, Bangladesh ABSTRACT Nowadays it is very important to maintain a high level security to ensure safe and trusted communication of information between various organizations. But secured data communication over internet and any other network is always under threat of intrusions and misuses. So Intrusion Detection Systems have become a needful component in terms of computer and network security. There are various approaches being utilized in intrusion detections, but unfortunately any of the systems so far is not completely flawless. So, the quest of betterment continues.

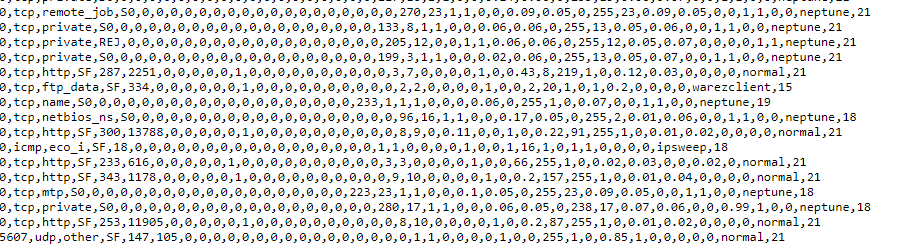
In this progression, here we present an Intrusion Detection System (IDS), by applying genetic algorithm (GA) to efficiently detect various types of network intrusions. Parameters and evolution processes for GA are discussed in details and implemented. This approach uses evolution theory to information evolution in order to filter the traffic data and thus reduce the complexity. To implement and measure the performance of our system we used the KDD99 benchmark dataset and obtained reasonable detection rate

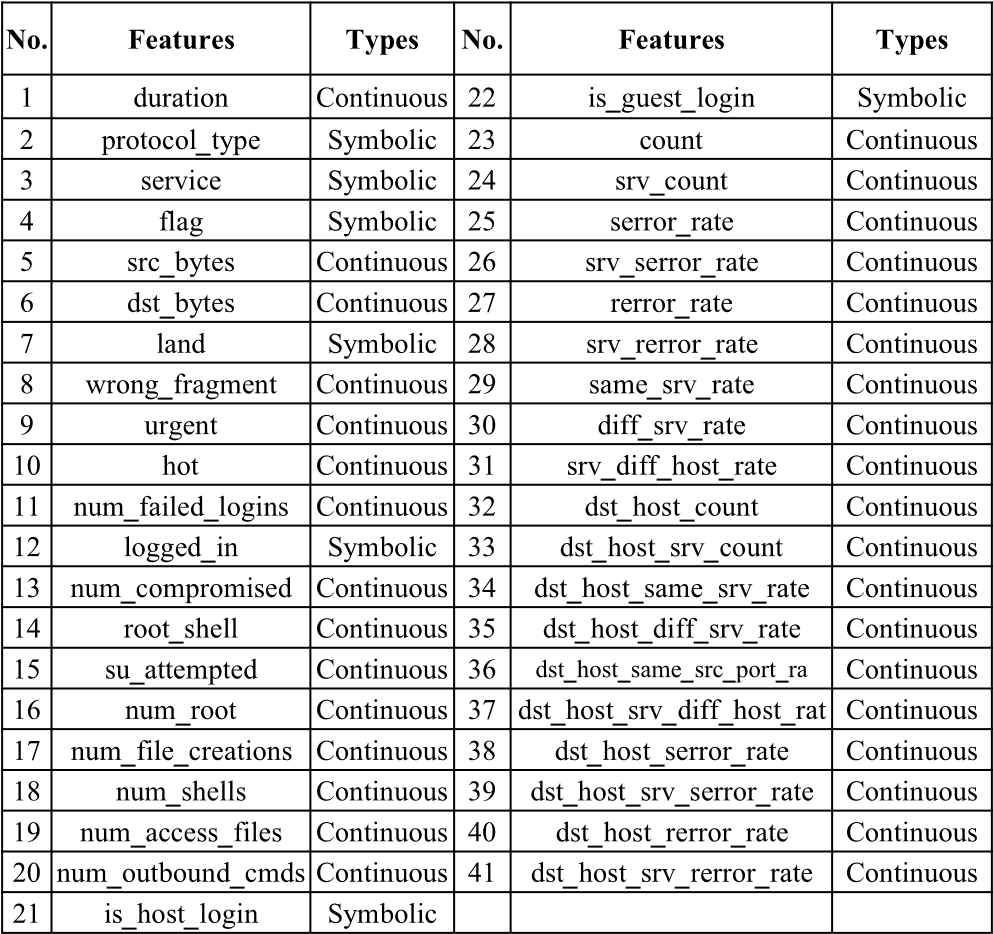
**DATASET DESCRIPTION**

The NSL-KDD dataset generated is widely used in intrusion detection experiments. In the latest literature [23]–[25], all the researchers use the NSL-KDD as the benchmark dataset, which not only effectively solves the inherent redundant records problems of the KDD Cup 1999 dataset but also makes the number of records reasonable in the training set and testing set, in such a way that the classifier does not favour more frequent records. The dataset covers the KDDTrain+ dataset as the training set and KDDTest+ and KDDTest−21 datasets as the testing set, which has different

**METHOD OF DATASET CREATION**

The datasets required for this work has been majorly collected from the “git hub Datasets”. Which are related to the network data. KDD datsets are fetched from google.





# Background Techniques

# RNN

# The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is [finite impulse](https://en.wikipedia.org/wiki/Finite_impulse_response) and the other is [infinite impulse](https://en.wikipedia.org/wiki/Infinite_impulse_response). Both classes of networks exhibit temporal [dynamic behavior](https://en.wikipedia.org/wiki/Dynamic_system).[[4]](https://en.wikipedia.org/wiki/Recurrent_neural_network#cite_note-4) A finite impulse recurrent network is a [directed acyclic graph](https://en.wikipedia.org/wiki/Directed_acyclic_graph) that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a [directed cyclic graph](https://en.wikipedia.org/wiki/Directed_cyclic_graph) that can not be unrolled

**Software Requirement Specification**

**4.1 Purpose:**

The objective of the pre-feasibility study is primarily to facilitate potential entrepreneurs in project identification for investment. The project pre-feasibility may form the basis of an important investment decision and in order to serve this objective, the document/study covers various aspects of project concept development, start-up, and production, marketing, finance and business management.

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

* 1. **Scope:**

The purpose of this document is to facilitate potential investors in intrusion prediction Service by providing them with a general understanding of the business with the intention of supporting potential investors in crucial investment decisions. The need to come up with pre-feasibility reports for undocumented or minimally documented sectors attains greater imminence as the research that precedes such reports reveal certain thumb rules; best practices developed by existing enterprises by trial and error, and certain industrial norms that become a guiding source regarding various aspects of business set-up and it’s successful management.

**Feasibility Study:**

This feasibility study analyses the market dynamics and financials of instrusion detection service, which is proposed for bigger cities across the country. The proposed prediction is to make future prediction based on current datasets so the users or the investors can check future prediction and invest.

**4.3.3.1 Operational Feasibility:**

The ““Intrusion Detection”” provides the good operational feasibility like the application should contain the rich set of operations like functionality that to understand by the user easily.it helps to find the resources very easily and more effective manner.

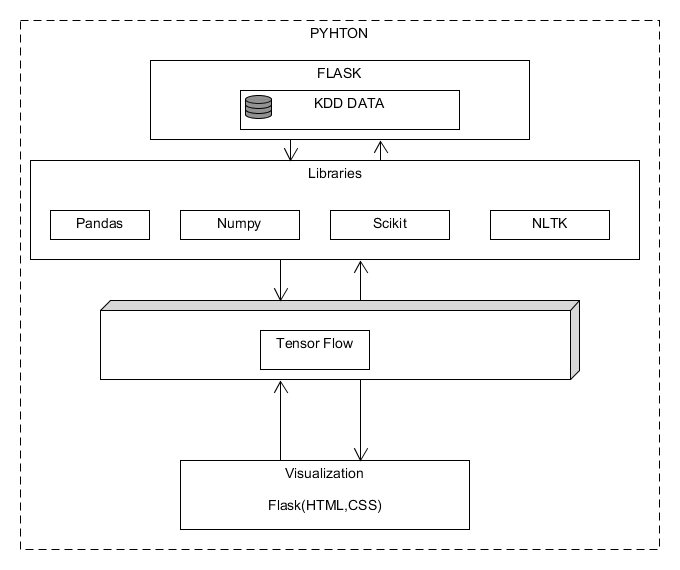
**4.3.3.2 Economic Feasibility:**

The main aim of the “Intrusion Detection” is to make the product in very low cost so that the everyone can understand and invest easily.

**4.3.3.3 Motivational feasibility:**

The motivational feasibility provides the effective user interface and helps the developer to motivate. There are many stakeholders of the system which helps the user to flow of the application according to they need. The stake holders provide the details to the developer in an easy understanding way.

**System Architecture:**



**Python:**

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects.[[27]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-7-27)

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library)

**Scikit Learn**

Scikit-learn (formerly scikits.learn and also known as sklearn) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language. It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy). Scikit-learn is largely written in Python, and uses [numpy](https://en.wikipedia.org/wiki/Numpy" \o "Numpy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) to improve performance. Support vector machines are implemented by a Cython wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR). In such cases, extending these methods with Python may not be possible.

**Tensor flow:**

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

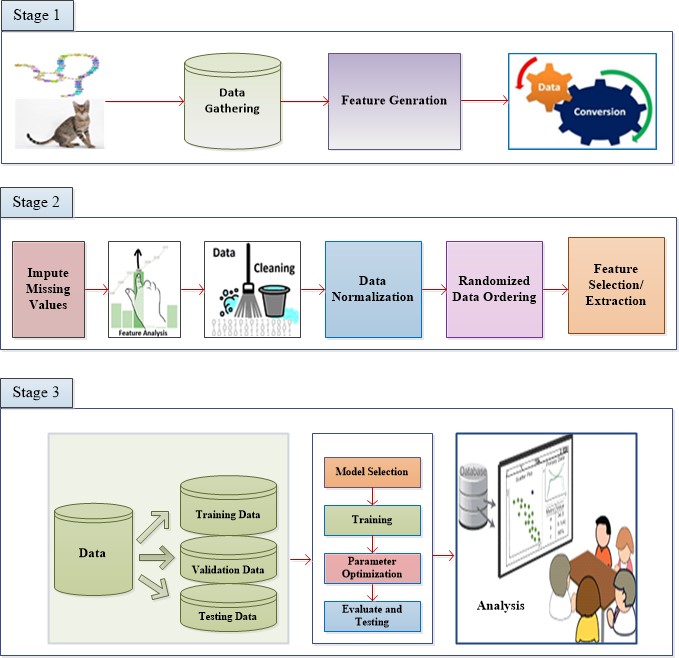
**Numpy**

NumPy or sometimes [/ˈnʌmpi/](https://en.wikipedia.org/wiki/Help:IPA/English) ([*NUM-pee*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key))) is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

**Pandas**

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), pandas is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license).[[2]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-2) The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals

**Archietecture:**



**Stage1:**

There are 41 features and 1 class label for every traffic record, and the features include basic features (No.1- No.10), content features (No.11 - No.22), and traffic features (No.23 - No.41) as shown in Table. According to their characteristics, attacks in the dataset are categorized into four attack types: DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack), and Probe (Probing attacks). The testing set has some specific attack types that disappear in the training set, which allows it to provide a more realistic theoretical basis for intrusion detection.

**Stage2:**

**Data Cleaning:**  
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

**Missing Data:**  
This situation arises when some data is missing in the data. It can be handled in various ways.  
Some of them are:

1. **Ignore the tuples:**  
   This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
2. **Fill the Missing values:**  
   There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value

**Stage 3:**

The obtained data from stage is taken into consideration then data is trained using the RNN neural network and obtained result is analysed and Showed in the graph using python library.

**PRODUCT FUNCTION:**

* Collected datasets of Intrusion detection from github from KDD datasest
* Pre-processing of obtained datasets
* Select Attributes which helps in predicting the Intrusion detection
* The selected datasets are trained using RNN
* The trained data sets are tested for Accuracy
* The obtained result is showed in the graph

**General Constraints**

The results generated have to be entered in to the system and any error or any value entered out of the boundary will not be understood by the system. In any case if the database crashes, the whole information collected and the results generated will be of no use.

**Specific Requirements**

**External Interface Requirements**

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface

**Hardware Requirements**

Processor : Intel i3 3.30 GHz.

Hard Disk : 40 GB (min)

Ram : 4GB

**Software Requirements**

Operating system : Windows 7 and above.

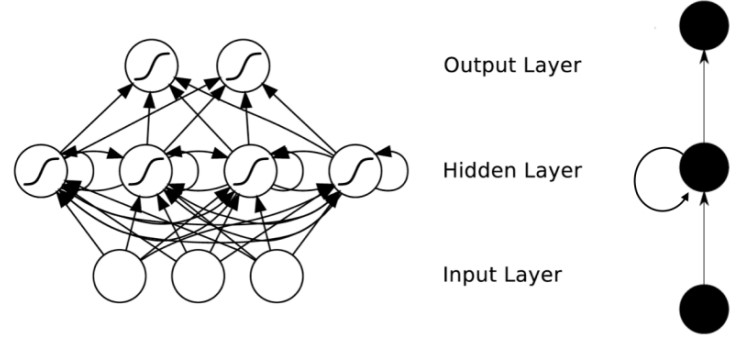
Coding Language : Html, CSS, Python.

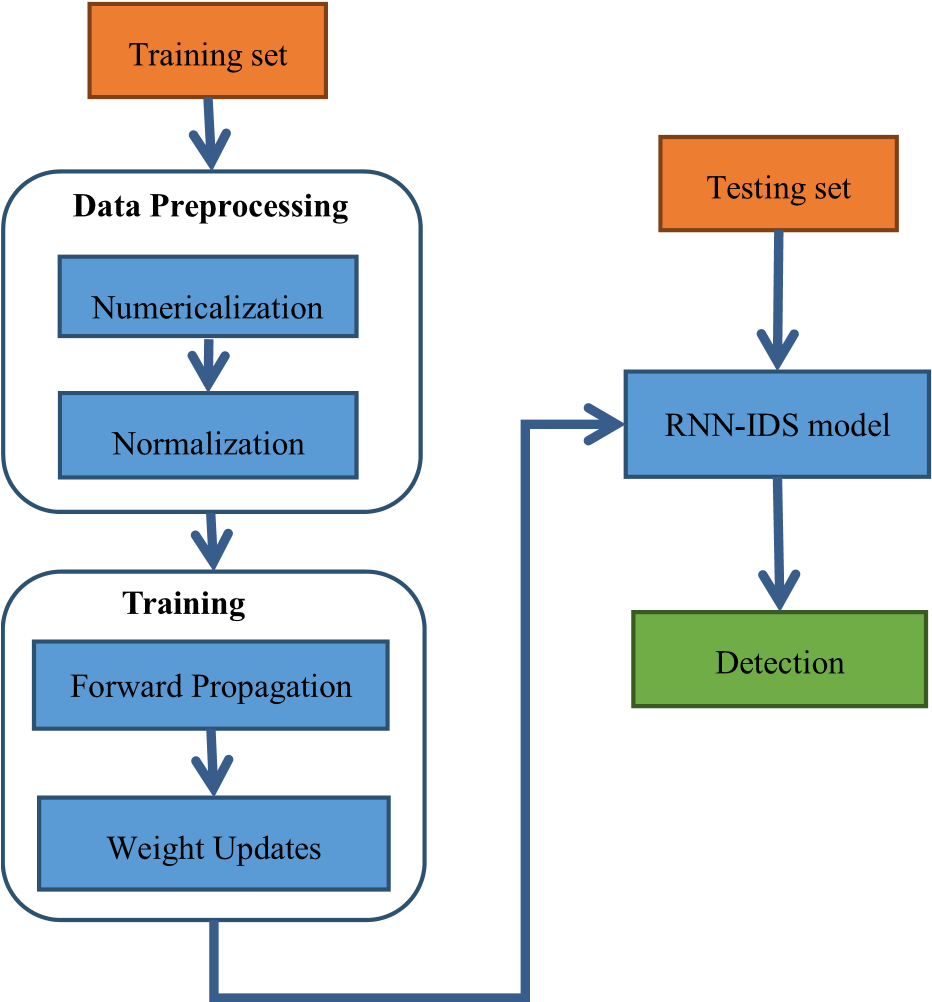
Framework: Flask

**RNN:**

Recurrent neural networks include input units, output units and hidden units, and the hidden unit completes the most important work. The RNN model essentially has a one-way flow of information from the input units to the hidden units, and the synthesis of the one-way information flow from the previous temporal concealment unit to the current timing hiding unit is shown in Fig. 1. We can regard hidden units as the storage of the whole network, which remember the endto-end information. When we unfold the RNN, we can find that it embodies the deep learning. A RNNs approach can be used for supervised classification learning.

Recurrent neural networks have introduced a directional loop that can memorize the previous information and apply

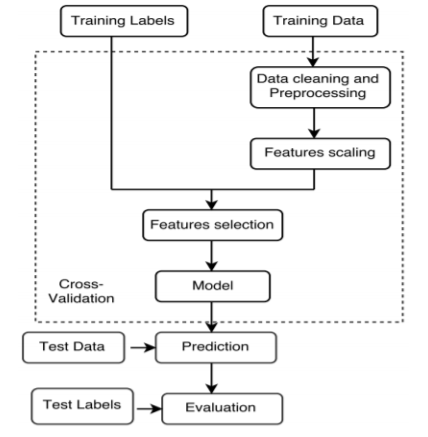




**RNN Block Diagram**

it to the current output, which is the essential difference from traditional Feed-forward Neural Networks (FNNs). The preceding output is also related to the current output of a sequence, and the nodes between the hidden layers are no longer connectionless; instead, they have connections. Not only the output of the input layer but also the output of the last hidden layer acts on the input of the hidden layer. The step involved in RNN-IDS

**DATAFLOW Diagram:**

****

Step 1: load datasets

load intrusion datasets that’s may be text file or csv file

Step 2: split datasets

split the datasets as training datasets and testing data sets

splitting ration is 0.75

Steps 3: data sets field names

assign field names(column name) to the datasets

column names :

Step 4: remove fields

remove the unwanted data, fields, columns from test datasets and train datasets

Step 5:mapping

assign class name to the different types of attack

example:

ipsweep=probe, teardrop=DoS , perl=U2R, Ftp\_write=R2L

and naormal=normal

* Step 6: Attack class distribution

**General Constraints**

The results generated have to be entered in to the system and any error or any value entered out of the boundary will not be understood by the system. In any case if the database crashes, the whole information collected and the results generated will be of no use.

**Non-functional requirements**

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirement.

**1. Accessibility**: It refers to the design of products, devices, services, or environments for people who experience disabilities. The concept of accessible design and practice of accessible development ensure both "direct access" (i.e. unassisted) and "indirect access" meaning compatibility with a person's assistive technology.

1. **Simplicity**: The project is driven by a simple user interface.
2. **Availability**: Availability of a system may also be increased by the strategy of focusing on increasing testability, diagnostics and maintainability and not on reliability. Improving maintainability during the early design phase is generally easier than reliability (and Testability & diagnostics). Maintainability estimates (item Repair by replacement rates) are also generally more accurate.

4. **Reliability**: The system should not crash and should identify invalid input and produce suitable error message.

5. **Usability:** The interface should be intuitive and easily navigable and user friendly.

6. **Integrity:** The software does not store any cache data or doesn‟t use system resources in background.

7. **Authentication**: Only authorized nodes can communicate with others.

### **SYSTEM DESIGN**

**4.1 Introduction to Design document**

The Software Design will be used to aid in software development for android application by providing the details for how the application should built. Within the Software Design, specifications are narrative and graphical documentation of the software design for the project includes use case models, sequence diagrams and other supporting requirement information.

**4.1.1 Scope**

This software Design Document is for a base level system, which will work as a proof of concept for the use of building a system that provides a base level of functionality to show feasibility for large-scale production use. The software Design Document, the focus placed on generation of the documents and modification of the documents. The system will used in conjunction with other pre-existing systems and will consist largely of a document interaction faced that abstracts document interactions and handling of the document objects. This Document provides the Design specifications of Cocoon.

**DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a way of representing a flow of data of a process or a system (usually an information system). The DFD also provides information about the output and inputs of each entity and the process itself. A data flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. There are several notations for displaying data flow diagrams.

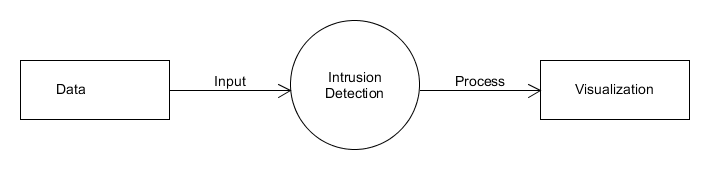
**Function**

**Database**

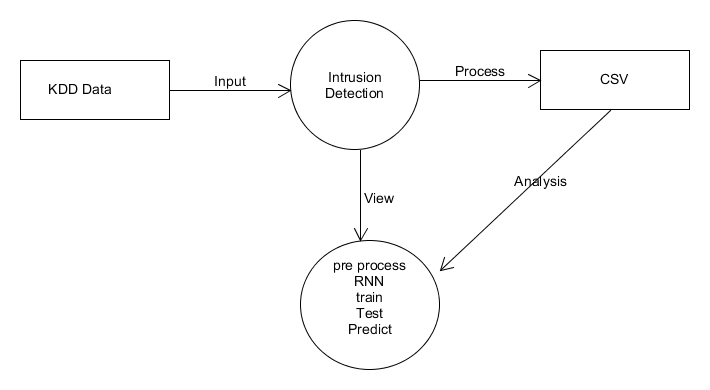
**Flow**

 **Input/Output**

**DFD: 0 Level**

****

**DFD Level 1:**

****

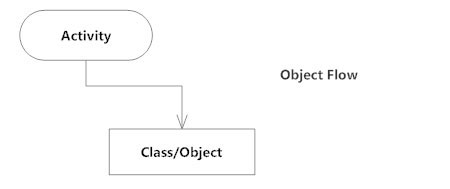
**ACTIVITY DIAGRAM**

An activity diagram visually presents a series of actions or flow of control in a system similar to a [flowchart](https://www.smartdraw.com/flowchart/) or a [data flow diagram](https://www.smartdraw.com/data-flow-diagram/). Activity diagrams are often used in business process modelling. They can also describe the steps in a [use case diagram](https://www.smartdraw.com/use-case-diagram/). Activities modelled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

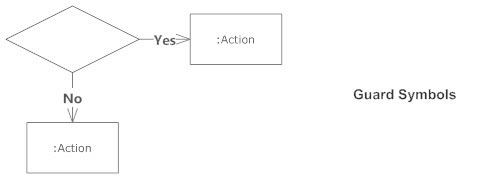
Start point symbol - Activity diagram

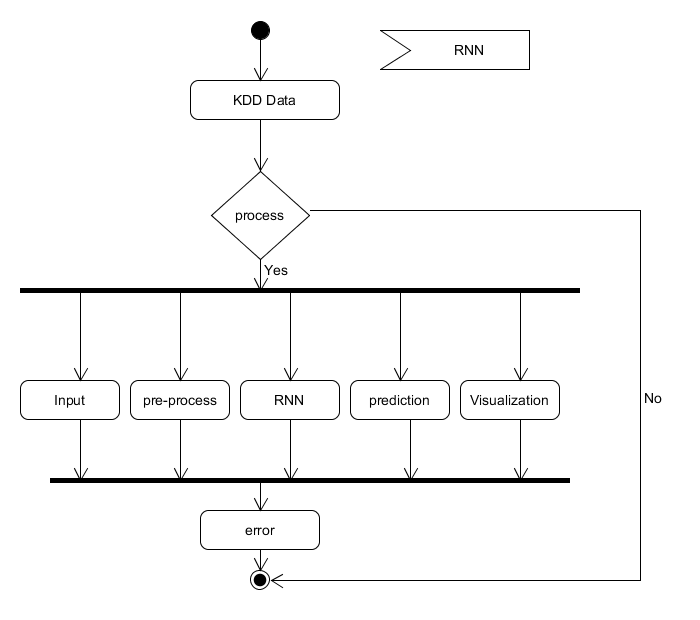


Action flow - Activity diagram



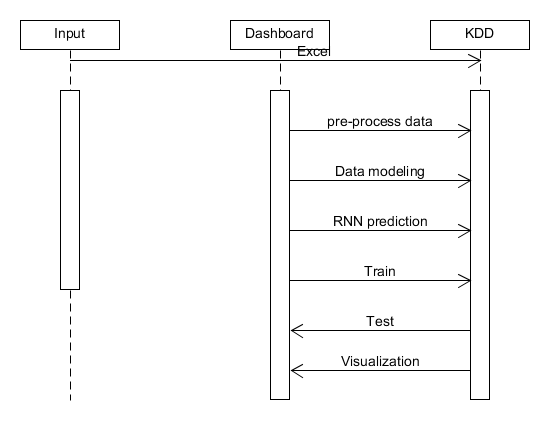




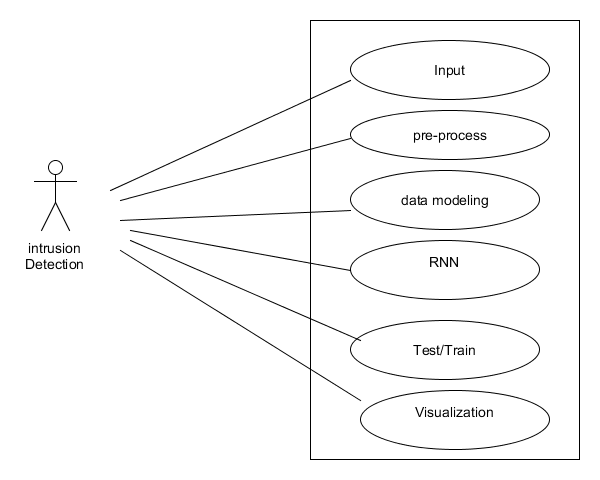
****

* 1. **SEQUENCE DIAGRAM**

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

****

**Use Case diagram:**

****

**Implementation**

The project is implemented using Python which is an object oriented programming language and procedure oriented programming language. Object oriented programming is an approach that provides a way of modularizing program by creating partitioned memory area of both data and function that can be used as a template for creating copies of such module on demand.

This project is implemented using python programming language. Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library). The machine Learning techniques are used in this project.

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

* The active user must be aware of the benefits of using the system
* Their confidence in the software built up
* Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

**User Training**

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important. Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

**Training on the Application Software**

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

**Operational Documentation**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

**System Maintenance**

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

**Corrective Maintenance**

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

**Adaptive Maintenance**

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

**Perceptive Maintenance**

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

**Preventive Maintenance**

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques

## Machine Learning vs Deep Learning

Experts in machine learning and deep learning have not yet reached consensus on these concepts. in this context, almost every day new ideas are being discussed. Machine Learning is an older concept than Deep Learning. Deep learning can also be called a technique that performs machine learning. The differences are listed below;

1. In deep learning, too much data is needed to bring the algorithm structure to the ideal.In machine learning, the problem can be solved with much less data because the person gives specific features to the algorithm.
2. Deep learning algorithms try to extract features from data. In machine learning, the features are determined by the expert.
3. While Deep Learning algorithms work on high performance machines, Machine Learning algorithms can work on ordinary CPUs.
4. In machine learning, the problem is usually divided into pieces, these parts are solved one by one and then the solutions are formed as a result of the solutions. In deep learning, the problem is solved end-to-end.
5. It takes a long time to train deep learning algorithms.

**RNN:**

Recurrent neural networks (RNNs) are connectionist models that capture the dynamics of sequences via cycles in the network of nodes. Unlike standard feedforward neural networks, recurrent networks retain a state that can represent information from an arbitrarily long context window. Although recurrent neural networks have traditionally been difficult to train, and often contain millions of parameters, recent advances in network architectures, optimization techniques, and parallel computation have enabled successful large-scale learning with them. In recent years, systems based on long short-term memory (LSTM) and bidirectional (BRNN) architectures have demonstrated ground-breaking performance on tasks as varied as image captioning, language translation, and handwriting recognition.

INTRUSION AND INTRUSION DETECTION SYSTEM Intrusion: Cyber attack incidents are increasing with the increasing use of internet. Cyber attack is the virtual life of the bullying in normal life. In this attack person encounters such situations as harassment, threats and blackmail. The attack may be in the form of the capture of the persons’ passwords or psychological pressure. Intrusion Detection System:IntrusionDetectionSystemsare very important software or hardware security tools to remove threats that would otherwise occur when carrying information, to prevent unauthorized access or abuse, and to report attacks to those responsible for security [3]. Attack Detection was ﬁrst introduced in Computer security threat monitoring and surveillance” survey published in 1980. The reasons for the need for intrusion detection systems;

1) It detects attacks that cannot be prevented by other security mechanisms.

2) It responds to the analysis phase before the attack occurs.

3) It allows attack analysis, system repair and the attacking factors to be corrected. Advantages of intrusion detection systems early detection, detailed information collection, evidence quality. The weaknesses of the intrusion detection systems are as follows; packet

fragmentation and timing attacks, mixing of scan sequence, package hijacking. It is difﬁcult to understand that packets arriving on the computer are sent for attack purposes. A packet arriving in the system may be sent for routine communication or an attack. Detecting an attack requires a difﬁcult and intensive calculation. Intrusion detection systems are classiﬁed according to several different criteria. IDSs can be classiﬁed; the architectural structure, the type of system it protects, and the processing time of the data. According to their location there are two typesofintrusiondetectionsystems, Host-Based and NetworkBased [4]. Also IDSs can be classiﬁed according to their techniques; Signature-Based and Anomaly-Based.

• Host-Based IDS; server tries to detect attacks by listening to the trafﬁc, registration ﬁles, and transactions.

• Network-Based IDS; listening to all the trafﬁc directed to the network, recording the content of each data packet passing through the network, cutting off attacks when necessary and creating reports.

• Signature-Based IDS; is used to detect known attack types.

• Anomaly-Based IDS; is used to detect unseen attacks.

Intrusion Detection Approaches In Intrusion Detection systems; techniques have been developed for modeling the data and create tables by classifying the modeled data [3]. The most used of these techniques are:

• Statistical:The ﬁrst examples of systems are based on statistical measurements. Using these examples, a statistical model is created by examining user or system behavior. New intrusions are tried to be determined with the created statistical model. Some of the statistical methods used in intrusion detection are Principal Component Analysis, Chi-square distribution, Gaussian Mixture Distribution.

• Artiﬁcial Neural Networks; models the given data using graphs of artiﬁcial neurons. They associate their vectors with their own algorithms and create new data. It is an approach used to examine and learn the behavior of data in the system [5]. With a enhanced form of ANN some authors prefer the use of Deep Learning for its efﬁciency [6].

• Support Vector Machines; It is the most preferred method for intrusion detection systems. Used for selection of feature vector. Supper Vector Machines aims to distinguish between data from two classes in a most appropriate way with a feature vector. They used in many classiﬁcation problems such as face recognition systems, sound analysis.

• Data Mining; it is known as reaching information among large-scale data. Used to extract rules by ﬁnding the relationship between data and users. Fuzzy Logic based on fuzzy set theory.

• Rule-Based Systems: It is developed by people who specialize in a speciﬁc area. These people examine the system trafﬁc and form rules and attack detection is done in this way.

• Fuzzy Logic: It is based on thinking like human beings and it is aimed to process them by converting them into mathematical functions

**Methodology**

**Data Collection:**

Data collection is one of the important thing in our project. The right dataset must be provided to get robust results. Our data mainly consists of previous year or weeks stock prices. We will be taking and analysing data from Kaggle. After seeing the accuracy, we will use the data in our model.

**Data Pre-Processing:**

Human can understand any type of data but machine cant so its better to make the data more machine readable. Raw data is usually inconsistent or incomplete. Data pre-processing involves checking missing values, splitting the dataset etc.

**Training Model:**

Similar to feeding somethings, machine/model should also learn by feeding on data. The data set extracted will be used to train the model. The training model uses a raw set of data as the undefined dataset which is collected from the previous fiscal year and from the same dataset a refine view is presented which is seen as the desired output. For the refining of the dataset various algorithms are implemented to show the desired output.

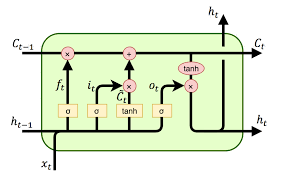
**System Evaluation:**

The dataset we use for the proposed project is taken from Kaggle. But this data set is in raw format. The data set is a collection of valuation of stock market information about some companies. The initial step is to convert raw data into processed data. Which is done by feature extraction, since the raw data collected have multiple attributes but only some of those attributes are needed for the prediction. Feature extraction is a reduction process. The structure, behaviour and views of a system is given by structural model.

**Long Short Term Memory:**

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

LSTMs have an edge over conventional feed-forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time. The purpose of this article is to explain LSTM and enable you to use it in real life problems.



In the case of LSTM architecture, the usual hidden layers are replaced with LSTM cells. The cells are composed of various gates that can control the input flow. An LSTM cell consists of input gate, cell state, forget gate, and output gate. It also consists of sigmoid layer, tanh layer and point wise multiplication operation.The various gates and their functions are as follows

* Input gate : Input gate consists of the input.
* Cell State : Runs through the entire network and has the ability to add or remove information with the help of gates.
* Forget gate layer: Decides the fraction of the information to be allowed.
* Output gate : It consists of the output generated by the LSTM.
* Sigmoid layer generates numbers between zero and one, describing how much of each component should be let through.
* Tanh layer generates a new vector, which will be added to the state.

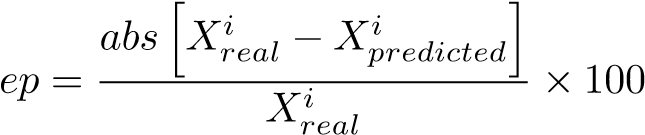
The cell state is updated based on the outputs form the gates. Mathematically we can represent it using the following equations.

|  |  |
| --- | --- |
| *ft* = *σ*(*Wf.*[*ht*−1*,xt*] + *bf*) | (2) |
| *it* = *σ*(*Wi.*[*ht*−1*,xt*] + *bi*) | (3) |
| *ct* = tanh(*Wc.*[*ht*−1*,xt*] + *bc*) | (4) |
| *ot* = *σ*(*Wo*[*ht*−1*,xt*] + *bo*) | (5) |
| *ht* = *ot* ∗ tanh(*ct*) | (6) |

where *xt*: input vector, *ht*: output vector, *ct*: cell state vector, *ft*: forget gate vector, *it*: input gate vector, *ot*: output gate vector and W,b are the parameter matrix and vector. Convolutional neural networks or CNNs, are a specialized kind of neural network for processing data that has a known, grid-like topology. This include time-series data, which can be thought of as a 1D and image data, which can be thought of as a 2D grid of pixels.The network employs a mathematical operation called convolution and hence known as convolutional neural network. It is a specialized kind of linear operation. Convolutional networks use convolution instead of

general matrix multiplication in at least one of their layers. The motivation behind using these three models is to identify whether there is any long term dependency existing in the given data. This can be identified from the performance of the models. RNN and LSTM architectures are capable of identifying long term dependencies and uses them for future prediction. However CNN architectures mainly focuses on the given input sequence and does not use any previous history or information during the learning process.The motivation behind testing the models with data from other companies is to check for interdependencies among the companies and to understand the market dynamics.

The train data was normalized. Test data was also subjected to the same normalization. After obtaining the predicted output, denormalization was applied and percentage error was calculated using the available true labels.The error percentage was calculated using (7)

 (7)

where ep is the error percentage,*Xreali* is the *ith* real value and *Xpredictedi* is the *ith* predicted value. Error percentage gives the magnitude of error present in the output.

**TESTING**

#### System Testing

System testing is the stage of implementation, which aimed at ensuring that the system works accurately and efficiently before the live operation commences. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successful achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to form, test can be conducted to demonstrate each function is fully operational .Knowing the internal working of a product, tests can be conducted to ensure that “all gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

#### 6.2 Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as „module testing‟. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

#### 6.3 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing, they are:

Top-down integration testing.

Bottom-up integration testing.

#### 6.4 Acceptance Testing

Acceptance testing or User Acceptance Testing (UAT) is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system‟s compliance with the business requirements and assess whether it is acceptable for delivery.

#### 6.5 Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Purpose** | **Test condition** | **Expected outcome** | **Actual result** | **Pass or Fail** |
| Load Data | Load instrusion data sets In CSV format. | If the data is not in the CSV format, shows a error message. | Load  datasets. | The data is loaded Successfully in CSV format. | Pass |
| Pre Process data | CSV data | If values are missing, or improper data | Pre processing is done | As  Expected. | Pass |
| RNN  Algorithm | Pre-processed data | KDD datasets  Trained for each available data | Training of data is complete | As  Expected. | Pass |
| RNN  Algorithm | Pre-processed data | KDD datasets  Trained for each available data | Testing of data is complete | As  Expected. | Pass |
| Prediction | Result obtained from RNN | Find networking Attacks | Attacks found | As  Expected. | Pass |

**Failed test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Purpose** | **Test condition** | **Expected outcome** | **Actual result** | **Pass or Fail** |
| Load Data | Load Automobile data sets In CSV format. | If the data is not in the CSV format, shows a error message. | Load  datasets. | The data is not loaded | fail |
| Pre Process data | CSV data | If values are missing, or improper data | Pre processing is not done | Result not found | fail |
| RNN  Algorithm | Pre-processed data | KDD datasets  Trained for each available data | Training of data is not complete | Error while Training, datas not found | Fail |
| RNN  Algorithm | Pre-processed data | KDD datasets  Trained for each available data | Testing of data is complete | Error while testing, datas not found | Fail |
| Prediction | Result obtained from RNN | Find networking Attacks | Attacks found | No Attacks or result error | Fail |

**RESULTS AND CONCLUSION:**

For preventing attacks to the networks, an intrusion de-

tection system plays a very critical role in the cybersecurity

domain. Its effectiveness directly depends on the used decision

For preventing attacks to the networks, an intrusion de-

tection system plays a very critical role in the cybersecurity

domain. Its effectiveness directly depends on the used decision

For preventing attacks to the networks, an intrusion detection system plays a very critical role in the cybersecurity domain. Its effectiveness directly depends on the used decision engine. To increase the flexibility of the system, instead of signature-based detection, it is required to implement the system as anomaly detection with a learning system. One of the newest training and classification technique, which is executed in this engine, is emerged as deep learning. Therefore, in this paper, it is aimed to provide a short survey of deep learning-based intrusion detection systems with the overview of various aspects of intrusion detection and deep learning algorithms. Additionally, this work lists and gives details about some publicly available datasets with their characteristics and shortcomings. We believe that this comprehensive survey on deep learning-based IDS could be helpful or researchers in this area. Although most of the researches proposed their system with the older dataset, as future work, it will be helpful to use the newest datasets with alternative deep learning approaches.

**Conclusion:**

In this project that is ” Deep Learning Approach for intrusion detection using recurrent neural network” we are analysing Intrusion detection attacks. For implementing this project, we have collected raw datasets from the kaggle.com. these datasets consist of 42 parameters like duration, protocol type, flag etc. by collecting all this dataset we are analysing the type of data and its class labels using recurrent neural network. Long short term memory (LSTM) is used for training and testing data. In this project 75% of datasets are trained and 25% of datasets are tested. The accuracy what we got after testing is shown with help of graph.

**Future Enhancement:**

In this project we identified 4 types of network attack type with 42 attributes. In the future enhancement we can apply for other network attack type and we can apply other deep learning algorithm for predicting the better result.

**Refernces:**

[1] Network Intrusion Detection Using Deep Neural Networks M.Ponkarthika1 and Dr.V.R.Saraswathy2 (Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 665-673, April-June 2018

[2] Host Based Intrusion Detection System with Combined CNN/RNN ModelAshima Chawla(B), Brian Lee, Sheila Fallon, and Paul Jacob

[3] On the Effectiveness of Machine and Deep Learning for Cyber Security 2018 10th International Conference on Cyber Conflict

[4] Collective Anomaly Detection Based on Long Short-Term Memory Recurrent Neural Networks Lo¨ıc Bontemps, Van Loi Cao(B), James McDermott, and Nhien-An Le-Khac

[5] A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection Anna L. Buczak, Member, IEEE, and Erhan Guven, Member, IEEE

[6] Application of Neural Networks for Intrusion Detection in Tor Networks Taro Ishitaki∗, Donald Elmazi†, Yi Liu ∗, Tetsuya Oda ∗, Leonard Barolli‡ and Kazunori Uchida‡2015 29th International Conference on Advanced Information Networking and Applications Workshops

[7] ApplicationofDeepRecurrentNeuralNetworksforPredictionofUserBehaviorin TorNetworks 2017 31st International Conference on Advanced Information Networking and Applications Workshops

[8] Deep Learning based Multi‐channel intelligent attack  detection for Data Security

Feng Jiang, Yunsheng Fu\* , B. B. Gupta

[9] Method of Intrusion Detection using Deep Neural Network Jin Kim, Nara Shin, Seung Yeon Jo and Sang

[10] Deep Learning based Attribute Classification Insider Threat Detection for Data Security

Fanzhi Meng, Fang Lou, Yunsheng Fu 2018 IEEE Third International Conference on Data Science in Cyberspac

[11] Concept for network intrusion detection system based on recurrent neural network classifier

[12] Rule E’xtraction from Neural Networks for Intrusion Detection in Computer Networks\*

[13] A Deep Learning Approach to Network Intrusion Detection Nathan Shone , Tran Nguyen Ngoc, Vu Dinh Phai

[14] DeepDefense: Identifying DDoS Attack via Deep Learning Xiaoyong Yuan∗, Chuanhuang Li