A MACHINE LEARNING BASED CYBER ATTACK DETECTION MODEL FOR WIRELESS SENSOR NETWORKS IN MICROGRIDS

A project report submitted in partial fulfillment of the requirements for

the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted By

S.Y. SAPTHAGIRI

(20UCA132)

Under the Guidance of

MS.R.B.SOWMYA.M.Sc.,

K.S. RANGASAMY COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)

NAAC Re-accredited and an ISO Certified Institution affiliated to Periyar University, Salem,

Included under 2(f) & 12B of UGC Act, 1956

K.S.R. Kalvi Nagar, Tiruchengode – 637 215

Namakkal District, Tamil Nadu, India

JUNE 2023

K.S. RANGASAMY COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)

NAAC Re-Accredited and an ISO Certified Institution affiliated to Periyar University, Salem,

Included under 2(f) & 12B of UGC Act, 1956

K.S.R. Kalvi Nagar, Tiruchengode – 637 215

Namakkal District, Tamil Nadu, India

A MACHINE LEARNING BASED CYBER ATTACK DETECTION MODEL FOR WIRELESS SENSOR NETWORKS IN MICROGRIDS

Bonafide work done by

S.Y. SAPTHAGIRI

(20UCA132)

A project report submitted in partial fulfillment of

the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted for the Viva-Voce Examination held on __________________

Signature of the Guide **Head of the Department**

Internal Examiner **External Examiner**

DECLARATION

DECLARATION

We here by declare that this project entitled **“A MACHINE LEARNING BASED CYBER ATTACK DETECTION MODEL FOR WIRELESS SENSOR NETWORKS IN MICROGRIDS ”** submitted to **K.S.Rangasamy College of Arts and Science (Autonomous), Tiruchengode – 637 215, Periyar University, Salem** is a record of original work done by ourself under the guidance of **MS.R.B.SOWMYA.M.Sc.,** Assistant Professor, Department of Computer Applications, K.S.Rangasamy College of Arts and Science (Autonomous), Tiruchengode and this project work has not formed the basis for the award of any Degree / Diploma / Associateship / Fellowship or similar title to any candidate of any university.

Place: Tiruchengode

Signature of the Candidate

Date :

S.Y. SAPTHAGIRI
(20UCA132)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am very much grateful to the almighty god and my parents who helped me all the way and who have modeled me into what I am today.

I acknowledge my sincere thanks to our chairman Dr. K.S. Rangasamy MJF and our Principal Dr.V.Radhakrishnan, K.S.Rangasamy College of Arts and Science, for their immense help in carrying out the project.

I pay my gratitude to our respectable Head of the Department Mr.T.S.Venkateswaran for enabling us to make use of the laboratory and library facility liberally which helped me a long way in carrying out my project work successfully.

I express my sincere thanks to our Project Coordinator Ms.S.Uma Parameshwari MCA, M.Phil, Assistant Professor and Department of Computer Applications, who has given valuable suggestions in reviews and coordinated us to complete this project.

I extend my sincere thanks to my Guide MS.R.B.SOWMYA.M.Sc., Assistant Professor and Department of Computer Applications, who guided me in the right way to finish my project without any trouble.

I sincerely thank all the staff members of the Department of Computer Applications for their kind co-operation during the period of my project work.

CONTENTS

TABLE OF CONTENT

SYNOPSIS

SYNOPSIS

With the development of the Internet, cyber-attacks are changing rapidly and the cyber security situation is not optimistic. Machine Learning (ML) and Deep Learning (DL) methods for network analysis of intrusion detection and provides a brief tutorial description of each ML/DL method. Papers representing each method were indexed, read, and summarized based on their temporal or thermal correlations. Because data are so important in ML/DL methods, they describe some of the commonly used network datasets used in ML/DL, discuss the challenges of using ML/DL for cyber security and provide suggestions for research directions.

The KDD data set is a well-known benchmark in the research of Intrusion Detection techniques. A lot of work is going on for the improvement of intrusion detection strategies while the research on the data used for training and testing the detection model is equally of prime concern because better data quality can improve offline intrusion detection.

This project presents the analysis of KDD data set with respect to four classes which are Basic, Content, Traffic and Host in which all data attributes can be categorized using PCA WITH MODIFIED RANDOM FOREST(MRF) and SVM. The analysis is done with respect to two prominent evaluation metrics, Detection Rate (DR) and False Alarm Rate (FAR) for an Intrusion Detection System (IDS).

As a result of this empirical analysis on the data set, the contribution of each of four classes of attributes on DR and FAR is shown which can help enhance the suitability of data set to achieve maximum DR with minimum FAR.

The experimental results obtained showed the proposed method successfully bring 91% classification accuracy using only 12 selected features and 97% classification accuracy using 36 features, while all 42 training features achieved 98% classification accuracy.

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1ABOUT THE ORGANIZATION

K.S.Rangasamy College of Arts and Science (KSRCAS) was started in the academic year 1995-1996 with the approval of the Government of Tamil Nadu and the University Grants Commission and it is affiliated to the Periyar University, Salem. It is ISO certified and NAAC accredited.

KSRCAS is situated in a sylvan environment in as prowling campus with well-designed academic edifices consisting of the highly advanced computer centers, well-furnished spacious lecture rooms and comfortable conference halls, most modern library including the digital enclosure, home- like hostel facilities separately for boys and girls, pleasant guest house for the visiting dignitaries, a bank with ATM features and a clinic to attend to the health of the students.

VISION, We strive for nurturing the potential of students by designing and delivering current, relevant and creative learning inputs. This is to achieve excellence in academics and to create socially responsible citizens. We are committed to shape global leaders and entrepreneurs, who create sustainable and fulfilling environment to the society.

MISSION Design and deliver learning inputs that are on par with global standards. Interface with business organizations, universities, research institutions, and government and non - government organizations. Design current, relevant inputs to transform students into entrepreneurs, employable and socially responsible citizens. Promote innovation and research in various areas of basic sciences, life sciences, computer science and humanities by way of interfacing with various funding organizations, universities and other research institutions.

1.2 PROJECT DESCRIPTION

With the development of the Internet, cyber-attacks are changing rapidly and the cyber security situation is not optimistic. Machine Learning (ML) and Deep Learning (DL) methods for network analysis of intrusion detection and provides a brief tutorial description of each ML/DL method. Papers representing each method were indexed, read, and summarized based on their temporal or thermal correlations. Because data are so important in ML/DL methods, they describe some of the commonly used network datasets used in ML/DL, discuss the challenges of using ML/DL for cyber security and provide suggestions for research directions.

The KDD data set is a well-known benchmark in the research of Intrusion Detection techniques. A lot of work is going on for the improvement of intrusion detection strategies while the research on the data used for training and testing the detection model is equally of prime concern because better data quality can improve offline intrusion detection.

This project presents the analysis of KDD data set with respect to four classes which are Basic, Content, Traffic and Host in which all data attributes can be categorized using PCA WITH MODIFIED RANDOM FOREST(MRF) The analysis is done with respect to two prominent evaluation metrics, Detection Rate (DR) and False Alarm Rate (FAR) for an Intrusion Detection System (IDS).

As a result of this empirical analysis on the data set, the contribution of each of four classes of attributes on DR and FAR is shown which can help enhance the suitability of data set to achieve maximum DR with minimum FAR.

The experimental results obtained showed the proposed method successfully bring 91% classification accuracy using only 12 selected features and 97% classification accuracy using 36 features, while all 42 training features achieved 98% classification accuracy

1.3 CYBER SECURITY

An interruption detection system is programming that screens a solitary or a system of PCs for noxious exercises that are gone for taking or blue penciling data or debasing system conventions. Most procedure utilized as a part of the present interruption detection systems are not ready to manage the dynamic and complex nature of digital assaults on PC systems. Cyber Security depicts an engaged writing review of machine learning and information digging techniques for digital investigation in help of interruption detection. In view of the quantity of references or the pertinence of a rising strategy, papers speaking to every technique were distinguished, perused, and compressed. Digital informational indexes utilized as a part of machine learning and information digging are portrayed for digital security is displayed, and a few proposals on when to utilize a given technique are given.

The Machine learning, Data Mining techniques are portrayed, and also a few utilizations of every strategy to digital interruption detection issues. The many-sided quality of various machine learning and information mining calculations is talked about, and the paper gives an arrangement of examination criteria for machine learning and information mining techniques and an arrangement of proposals on the best strategies to utilize contingent upon the attributes of the digital Issue to tackle Cyber security is the arrangement of advances and procedures intended to ensure PCs, systems, projects, and information from assault, unapproved access, change, or pulverization. They are engaging a result of their capacity to recognize zero-day assaults.

1.4 INTRUSION DETECTION

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 An Intrusion Detection System is an application used for monitoring the network and protecting it from the intruder. With the rapid progress in the internet-based technology new application areas for computer network have emerged. In instances, the fields like business, financial, industry, security and healthcare sectors the LAN and WAN applications have progressed.it is the burglar alarm that detects that the lock has been broken and alerts the owner by raising an alarm. Moreover, Firewalls do a very good job of filtering the incoming traffic from the Internet to circumvent the firewall. For example, external users can connect to the Intranet by dialing through a modem installed in the private network of the organization. this kind of access cannot be detected by the firewall. An Intrusion Prevention System (IPS) is a network security/threat prevention technology that audits network traffic flows to detect and prevent vulnerability exploits. There are two types of prevention system they are Network Intrusion Prevention System (NIPS) and Host Intrusion Prevention System (HIPS). These systems watch the network traffic and automatically take actions to protect networks and systems. IPS issue is false positives and negatives. False positive is defined to be an event which produces an alarm in IDS where there is no attack. False negative is defined to be an event which does not produces an alarm when there is an attack takes place.

1.5 MACHINE LEARNING

Machine learning is one of the most exciting recent technologies in Artificial Intelligence. Learning algorithms in many applications that's they make use of daily. Every time a web search engine like Google or Bing is used to search the internet, one of the reasons that works so well is because a learning algorithm, one implemented by Google or Microsoft, has learned how to rank web pages. Spam filters in email saves the user from having to wade through tons of spam email, that's also a learning algorithm. Machine learning, a brief review and future prospect of the vast applications of machine learning has been made.

According to Arthur Samuel Machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. A more formal definition was given by Tom Mitchell as a computer program is said to learn from EXPERIENCE (E) with respect to some and TASK (T) some PERFORMANCE MEASURE (P), if its performance on T, as measured by P, improves with experience E then the program is called a machine learning program. And the performance measure P, was the probability that it won the next game of checkers against some new opponent. In all fields of engineering, there are larger and larger data sets that are being understood using learning algorithms.

1.6 SUPERVISED LEARNING

This learning process is based on the comparison of computed output and expected output, that is learning refers to computing the error and adjusting the error for achieving the expected output. For example, a data set of houses of particular size with actual prices is given, then the supervised algorithm is to produce more of these right answers such as for new house what would be the price.

1.7 UNSUPERVISED LEARNING

Unsupervised learning is termed as learned by its own by discovering and adopting, based on the input pattern. In this learning the data are divided into different clusters and hence the learning is called a clustering algorithm. One example where clustering is used is in Google News (URL news.google.com). Google News groups new stories on the web and puts them into collective news stories.

1.8 REINFORCEMENT LEARNING

Reinforcement learning is based on output with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward. A reward is given for correct output and a penalty for wrong output. Reinforcement learning differs from the supervised learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected.

RELATED WORK

CHAPTER-2

RELATED WORK

There are many puzzles about the relationship among ML, DL, and Artificial Intelligence (AI). AI is a new technological science that studies and develops theories, methods, techniques, and applications that simulate, expand and extend human intelligence. It is a branch of computer science that seeks to understand the essence of intelligence and to produce a new type of intelligent machine that responds Ina manner similar to human intelligence. Research in this area includes robotics, computer vision, nature language processing and expert systems. AI can simulate the information process of human consciousness, thinking. AI is not human intelligence, but thinking like a human might also exceed human intelligence. ML is a branch of AI and is closely related to (and often overlaps with) computational statistics, which also focuses on prediction making using computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. ML is occasionally co fixated with data mining, but the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. ML can also be unsupervised and be used to learn and establish baseline behavioral paroles for various entities and then used to meaningful anomalies.

The pioneer of ML, Arthur Samuel, defined ML as a “field of study that gives computers the ability to learn without being explicitly programmed". ML primarily focuses on classification and regression based on known features previously learned from the training data. DL is a new field in machine-learning research. Its motivation lies in the establishment of a neural network that simulates the human brain for analytical learning. It mimics the human brain mechanism to interpret data such as images, sounds and texts. The concept of DL was proposed by Hinton based on the Deep Belief Network (DBN), in which an unsupervised greedy layer-by-layer training algorithm is proposed that provides hope for solving the optimization problem of deep structure. Then the deep structure of a multi-layer automatic encoder is proposed. In addition, the convolution neural network proposed by is the first real multilayer structure learning algorithm that uses a space relative relationship to reduce the number of parameters to improve the training performance.

DL is a machine-learning method based on characterization of data learning. An observation, such as an image, can be expressed in a variety of ways, such as a vector of each pixel intensity value, or more abstractly as a series of edges, a region of a particular shape, or the like. Using specific representations makes it easier to learn tasks from instances. Similarly, to ML methods, DL methods also have supervised learning and unsupervised learning. Learning models built under different learning frameworks are quite different. The benefit of DL is the use of unsupervised or semi-supervised feature learning and hierarchical feature extraction to efficiently replace features manually.

The differences between ML and DL include the following:

- **Data dependencies.** The main difference between deep learning and traditional machine learning is its performance as the amount of data increases. Deep learning algorithms do not perform as well when the data volumes are small, because deep learning algorithms require a large amount of data to understand the data perfectly. Conversely, in this case, when the traditional machine learning algorithm uses the established rules, the performance will be better.
- **Hardware dependencies.** The DL algorithm requires many matrix operations. The GPU is largely used to optimize matrix operations efficiently. Therefore, the GPU is the hardware necessary for the DL to work properly. DL relies more on high-performance machines with GPUs than do traditional machine-learning algorithms.
- **Feature processing.** Feature processing is the process of putting domain knowledge into a feature extractor to reduce the complexity of the data and generate patterns that make learning algorithms work better. Feature processing is time consuming and requires specialized knowledge. In ML, most of the characteristics of an application must be determined by an expert and then encoded as a data type. Features can be pixel values, shapes, textures, locations, and orientations. The performance of most ML algorithms depends upon the accuracy of the features extracted. Trying to obtain high level features directly from data is a major difference between DL and traditional machine-learning algorithms Thus; DL reduces the effort of designing a feature extractor for each problem. Problem-solving method. When applying traditional machine-learning algorithms to solve problems, traditional machine learning usually breaks down the problem into multiple sub

problems and solves the sub-problems, ultimately obtaining the final result. In contrast, deep learning advocates direct end-to-end problem solving.

- **Execution time.** In general, it takes a long time to train a DL algorithm because there are many parameters in the DL algorithm; therefore, the training step takes longer. The most advanced DL algorithm, such as Reset, takes exactly two weeks to complete a training session, whereas ML training takes relatively little time, only seconds to hours. However, the test time is exactly the opposite. Deep learning algorithms require very little time to run during testing.

Compared with some ML algorithms, the test time increases as the amount of data increases. However, this point does not apply to all ML algorithms, because some ML algorithms have short test times Interpretability. Crucially, interpretability is an important factor in comparing ML with DL. DL recognition of handwritten numbers can approach the standards of people, a quite amazing performance. However, a DL algorithm will not tell why it provides this result. Of course, from a mathematical point of view, a node of a deep neural network is activated. However, how should neurons be modeled and how do these layers of neurons work together? Thus, it is difficult to explain how the result was generated. Conversely, the machine-learning algorithm provides explicit rules for why the algorithm chooses so; therefore, it is easy to explain the reasoning behind the decision.

The steps of a DL approach are similar to ML, but as mentioned above, unlike machine-learning methods; its feature extraction is automated rather than manual. Model selection is a constant trial and error process that requires a suitable ML/DL algorithm for different mission types. There are three types of ML/DL approaches: supervised, unsupervised and semi-supervised. In supervised learning, each instance consists of an input sample and a label. The supervised learning algorithm analyzes the training data and uses the results of the analysis to map new instances. Unsupervised learning is a machine-learning task that deduces the description of hidden structures from unlabeled data. Because the samples unlabeled, the accuracy of the algorithm's output cannot be evaluated, and only the key features of the data can be summarized and explained. Semi-supervised learning is a means of combining supervised learning with unsupervised learning. Semi-supervised learning uses a large amount of unlabeled data when using labeled data for pattern recognition using semi-supervised learning can reduce label efforts while achieving high accuracy.

Commonly used ML algorithms include for example K-Nearest Neighbor (KNN), MRF, Decision Tree, and Bayes. The DL model includes

For Example

- **Deep Belief Network** (DBM),
- **Convolutional Neural Network** (CNN),
- **Long-Short Term Memory** (LSTM).

There are many parameters such as the number of layers and nodes to choose, but also to improve the model and integration. After the training is complete, there are alternative models that must be evaluated on different aspects. The evaluation model is a very important part of the machine-learning mission. Different machine learning missions have various evaluation indicators, whereas the same types of machine-learning missions also have different evaluation indicators, each with a different emphasis such as classification, regression, clustering and the like. The confusion matrix is a table that describes the classification results in detail, whether they are correctly or incorrectly classified and different classes are distinguished, for a binary classification a 2 * 2 matrix, and for n classification, an n * n matrix.

TABLE 1. Confusion matrix

Matrix tables	**Predicted as positive**	**Predicted as negative**
Labeled as positive	True Positive (TP)	False Negative (FN)
Labeled as negative	False Positive (FP)	True Negative (TN)

- True Positive (TP): Positive samples correctly classified by the model;
- False Negative (FN): A positive sample that is misclassified by the model;

- False Positive (FP): A negative samples that is misclassified by the model;
- True Negative (TN): Negative samples correctly classified by the model;

Further, the following metrics can be calculated from the confusion matrix:

- Accuracy: (TP + TN)/ (TP + TN + FP + FN). Ratio of the number of correctly classified samples to the total number of samples for a given test data set. When classes are balanced, this is a good measure; if not, this metric is not very useful.
- Precision: TP/ (TP + FP). It calculates the ratio of all "correctly detected items" to all "actually detected items".
- Sensitivity or Recall or True Positive Rate (TPR): TP/ (TP + FN). It calculates the ratio of all "correctly detected items" to all "items that should be detected".
- False Negative Rate (FNR): FN/ (TP + FN). The ratio of the number of misclassified positive samples to the number of positive samples.
- False Positive Rate (FPR): FP/ (FP + TN). The ratio of the number of misclassified negative samples to the total number of negative samples.
- True Negative Rate (TNR): TN/ (TN + FN). The ratio of the number of correctly classified negative samples to the number of negative samples.
- F1-score:2*TP / (2*TP + FN + FP). It calculates the harmonic mean of the precision and the recall.
- ROC: In ROC space, the abscissa for each point is FPR and the ordinate is TPR, which also describes the trade-off of the classifier between TP and FP. ROC's main analysis tool is a curve drawn in ROC space - the ROC curve.

In the area of Cyber Security, the metrics commonly used in assessment models are precision, recall, and F1-score. The higher and better the precision and recall of model tests are, the better, but in fact these two are in some cases contradictory and can only be emphatically balanced according to the task needs. The F1-score is the harmonic average of precision and recall, considering their results. In general, the higher the F1-score, the better the model will perform.

SYSTEM ANALYSIS

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

- A new (emerging) topic is something people feel like discussing, commenting, or forwarding the information further to their friends. Conventional approaches for topic detection have mainly been concerned with the frequencies of (textual) words.
- Microgrid as a small-size power system covers both the generation and consumption sides which makes it possible to operate in two operation modes of grid-connected and islanded. Beside the physical layer distributed generations (DGs) and renewable energy sources, loads and storage units, a microgrid has an interconnected cyber layer mainly dealing with data transmission and decision making based on data gathering.
- It cannot be applied when the contents of the messages are mostly non textual information. On the other hand, the "words" formed by mentions are unique, require little preprocessing to obtain (the information is often separated from the contents), and are available regardless of the nature of the contents.

3.1.1 DRAWBACKS OF EXISTING SYSTEM:

- ✓ Detection of User Cluster with Suspicious Activity Group of users with suspicious activities has to be identified using anomaly detection.
- ✓ Approach to detect suspicious profiles on social platforms Aim of a dynamic approach is to alert the users of Smartphone users about suspicious profiles located in his or her close circle of contacts on a given social network.
- ✓ Detection of Random Link Attacks Malicious users create false identities and used it to communicate with innocent users.
- ✓ Threat Detection through Graph Learning and Psychological Context is very tedious.
- ✓ Low accuracy.

3.2 PROPOSED SYSTEM:

- In this project, we have proposed a new approach to detect the emergence of topics in a social network stream. The basic idea of our approach is to focus on the social aspect of the Grids reflected in the mentioning behavior of users instead of the textual contents.
- We have proposed a probability model that captures both the number of mentions per Grid and the frequency of mentioner.
- For each new Grid we use samples within the past T time interval for the corresponding user for training the mention model we propose below.
- We assign anomaly score to each Grid based on the learned probability distribution. The score is then aggregated over users and further fed into a change point analysis.
- The proposed methodology The Modified Random Forest method makes use of the feed-forward NN model to construct optimal PIs surrounding the forecast target. In order to detect data integrity attack in the smart sensors installed in the microgrid local consumers' side, each PI is in charge of modeling the forecast uncertainty existing in the electric consumption data. Each PI is made up of a lower bound and an upper bound such that any forecast sample will fall between these two bounds.
- It also to increase its adaptability and flexibility the studied parameter value selected automatically according to the used training dataset. And also decrease the detection generation time by enhancing the clustering.

3.2.1 ADVANTAGES PROPOSED SYSTEM:

- The proposed method does not rely on the textual contents of social network Grids, it is robust to rephrasing and it can be applied to the case where topics are concerned with information other than texts so on.
- The proposed link-anomaly-based methods performed even better than the keyword-based methods on "ICS Smart Grid" data sets.
- High in accuracy.
- Minimum computation time
- Fast and easily find anomaly users.

3.3 SYSTEM REQUIREMENT AND SPECIFICATION

3.3.1HARDWARE REQUIREMENTS:

➢ Processor Type	: AMD RYZEN 7
➢ Speed	: 4.40GHZ
➢ RAM	:16 GB RAM
➢ Hard disk	: 1 TB
➢ Keyboard	: 101/102 Standard Keys
➢ Mouse	: Optical Mouse

3.3.2 SOFTWARE REQUIREMENTS:

➢ Operating System	: Windows 10
➢ Front End	: Jupyter Notebook/ Anaconda tool
➢ Coding Language	: Python

3.4. FEATURES OF SOFTWARE

FRONT END: JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. The software requirement specification is created at the end of the analysis task. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

FEATURES OF JUPYTER NOTEBOOK

- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.
- The ability to execute code from the browser, with the results of computations attached to the code which generated them.
- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc.
- For example, publication-quality figures rendered by the matplotlib library, can be included inline.
- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.
- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MATRIX.

NOTEBOOK DOCUMENTS

Notebook documents contains the inputs and outputs of a interactive session as well as additional text that accompanies the code but is not meant for execution.

In this way, notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics, and rich representations of resulting objects. These documents are internally JSON files and are saved with the. ***ipynb** extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues. Notebooks may be exported to a range of static formats, including HTML (for example, for blog posts), Re-Structured Text, LaTeX, PDF, and slide shows, via the unconvert command.

Furthermore, any. ipynb notebook document available from a public URL can be shared via the Jupyter Notebook Viewer (ipynb viewer). This service loads the notebook document from the URL and renders it as a static web page. The results may thus be shared with a colleague, or as a public blog post, without other users needing to install the Jupyter notebook themselves. In effect, ipynb viewer is simply unconvert as a web service, so you can do your own static conversions with ipynb convert, without relying on ipynb viewer.

PYTHON DEFINITION

Python is a high-level programming language designed to be easy to read and simple to implement. It is open source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.

Python is considered a scripting language, like Ruby or Perl and is often used for creating Web applications and dynamic Web content. It is also supported by a number of 2D and 3D imaging programs, enabling users to create custom plug-ins and extensions with Python. Examples of applications that support a Python API include GIMP, Inkscape, Blender, and Autodesk Maya.

Scripts written in Python (.PY files) can be parsed and run immediately. They can also be saved as a compiled programs (.PYC files), which are often used as programming modules that can be referenced by other Python programs.

PYTHON FEATURES

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

1) **Easy to Learn and Use**

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) **Expressive Language**

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print ("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

3) **Interpreted Language**

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) **Cross-platform Language**

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) **Free and Open Source**

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

6) **Object-Oriented Language**

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

7) **Extensible**

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

8) **Large Standard Library**

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, NumPy, Karas, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

9) **GUI Programming Support**

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

10) **Integrated**

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++ Java. It makes easy to debug the code.

11. **Embeddable**

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

12. **Dynamic Memory Allocation**

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to x, then we don't need to write int x = 15. Just write x = 15.

ANACONDA

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, ANACONDA and packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using. Cloud hosts hundreds of useful Python packages, notebooks, projects and environments for a wide variety of applications. You do not need to log in, or even to have a Cloud account, to search for public packages, download and install them.

You can build new anaconda packages using anaconda-build, then upload the packages to Cloud to quickly share with others or access yourself from anywhere. The Anaconda Cloud Command Line Interface (CLI), anaconda-client, allows you to manage your account - including authentication, tokens, upload, download, remove and search. Connect to and manage your Anaconda Cloud account. Upload packages you have created. Generate access tokens to allow access to private packages.

For developers, Cloud is designed to make software development, release and maintenance easy by providing broad package management support. Cloud allows for free public package hosting, as well as package channels, providing a flexible and scalable service for groups and organizations of all sizes.

3.4.1 APPLICATIONS PROVIDED IN ANACONDA DISTRIBUTION

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

- JupyterLab
- Jupyter Notebook
- Qt Console
- Spyder
- Glue viz
- Orange3
- RStudio
- Visual Studio Code

Jupyter Lab: This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Jupyter Notebook: This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.

Qt Console: It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.

Spyder: Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.

VS Code: It is a streamlined code editor with support for development operations like debugging, task running and version control.

Glue viz: This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.

Orange 3: It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.

RStudio: It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks

SYSTEM DESIGN

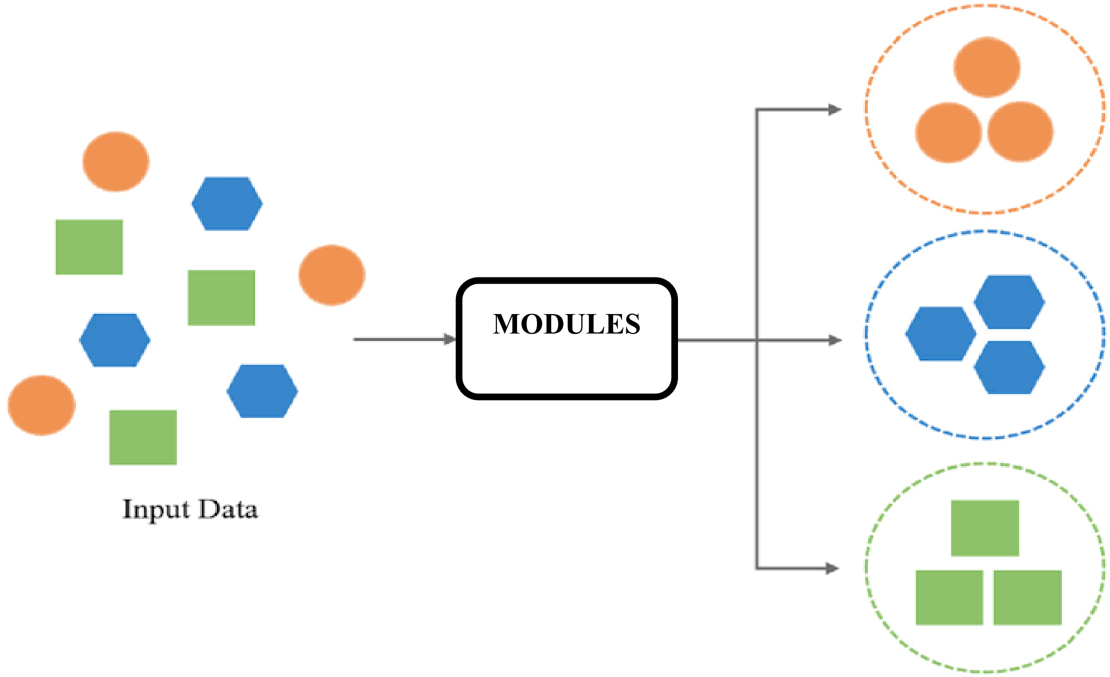
AND DEVELOPMENT

CHAPTER- 4

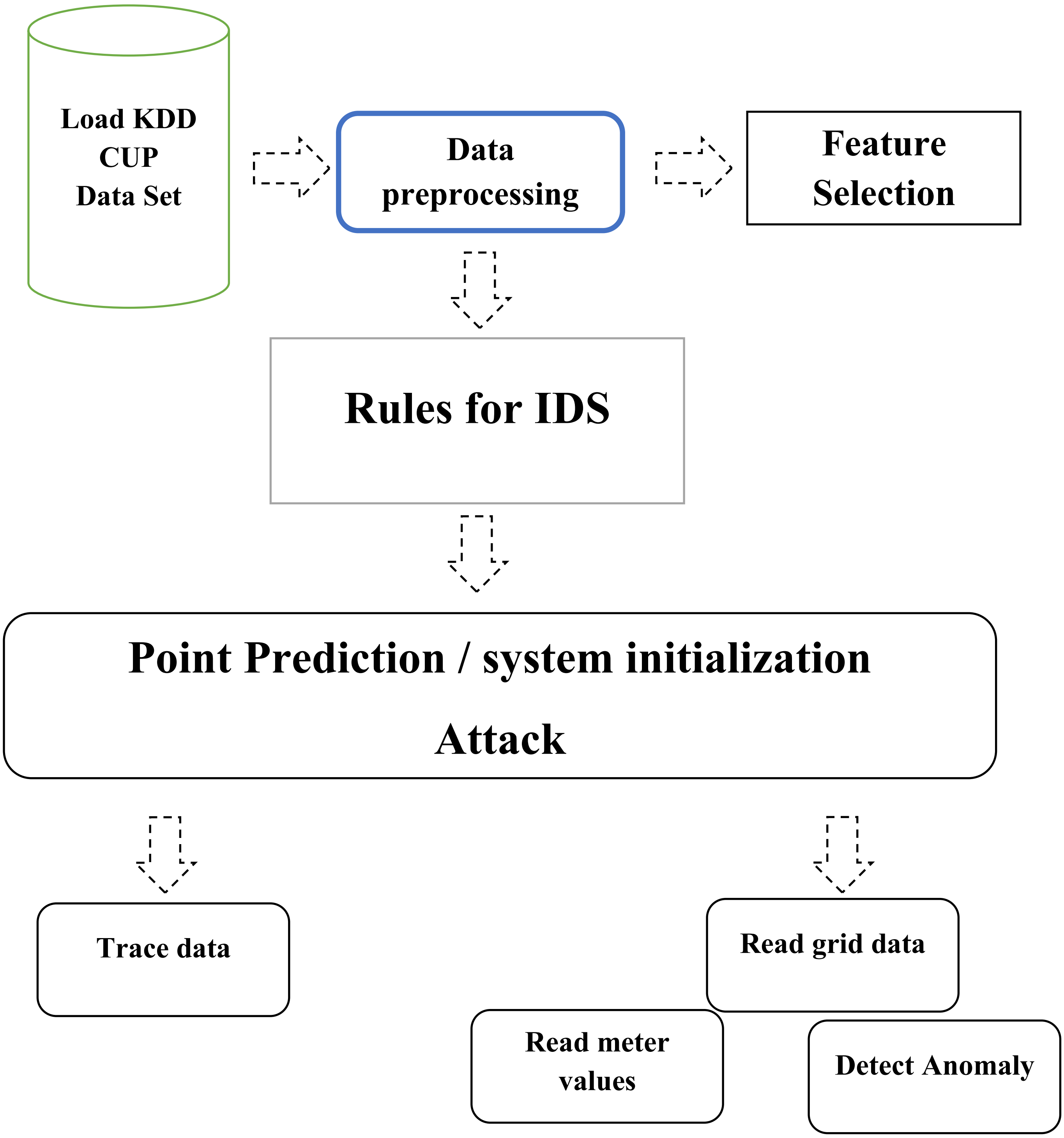
4.1 SYSTEM DESIGN AND DEVELOPMENT

4.1.1 SYSTEM FLOW DIAGRAM

Data constitute the basis of computer network security research. The correct choice and reasonable use of data are the prerequisites for conducting relevant security research. The size of the dataset also affects the training effects of the ML and DL models.



4.2. DATA FLOW DIAGRAM



4.3MODULES DESCRIPTION

4.3.1 DATA PREPROCESSING:

In this module, we preprocess the probability model that we used to capture the normal mentioning behavior of a user and how to train the model. We characterize a Grid in a social network stream by the number of mentions k it contains, and the set V of names (IDs) of the mentioned (users who are mentioned in the Grid). To detect data integrity attack in the microgrid. According to this module, we pre-process the numerical values read from microgrid constructing PIs around the smart meter readings of the electric consumers can determine the normal or abnormal behaviors in the system. In the case of cyber security, the proposed anomaly detection method may make any of these four decisions: 1) true positive, 2) false positive, 3) true negative and 4) false negative. These decisions are made depending on the real system data and the proposed anomaly detection model response. The PIs created by the proposed LUBE based method will make boundaries which will help detecting anomalies.

4.3.2 COMPUTING THE LINK-ANOMALY SCORE:

In this module, we describe how to compute the deviation of a user's behavior from the Norma mentioning behavior modeled In order to compute the anomaly score of a new Grid by user actual time containing mentions to users reading the grid value We compute the probability with the training set T (t) u, which is the collection of Grids by user u in the time period [t−T, t] (we use T = 30 days in this project). Accordingly, the link-anomaly score is defined the two terms in the above equation can be computed via the predictive distribution of the number of mentions, and the predictive distribution of the mentioned.

4.3.3 CHANGE POINT ANALYSIS AND DTO:

This technique is an extension of Change Finder proposed, that detects a change in the statistical dependence structure of a time series by monitoring the compressibility of a new piece of data. This module is to use a Modified Random Forest (NML) coding called MRF coding as a coding criterion instead of the plug-in predictive distribution used. Specifically, a change point is detected through two layers of scoring processes. The first layer detects outliers and the second layer detects change-points.

4.3.4 MODIFIED RANDOM FOREST DETECTION METHOD:

In this module that to the change-point detection based on MRF followed by DTO described in previous sections, we also test the combination of our method with Kleinberg's Modified Random Forest-detection method. More specifically, we implemented a two-state version of Kleinberg's Modified Random Forest-detection model. The reason we chose the two-state version was because in this experiment we expect nonhierarchical structure. The Modified Random Forest-detection method is based on a probabilistic automaton model with two states, Modified Random Forest state and non-Modified Random Forest state. Some events (e.g., arrival of Grids) are assumed to happen according to a time-varying Poisson processes whose rate parameter depends on the current state.

4.4 SYSTEM DESIGN

Data constitute the basis of computer network security research. The correct choice and reasonable use of data are the prerequisites for conducting relevant security research. The size of the dataset also affects the training effects of the ML and DL models. Computer network security data can usually be obtained in two ways: 1) directly and 2) using an existing public dataset. Direct access is the use of various means of direct collection of the required cyber data, such as through Win Dump or Wire shark software tools to capture network packets. This approach is highly targeted and suitable for collecting short-term or small amounts of data, but for long-term or large amounts of data, acquisition time and storage costs will escalate. The use of existing network security datasets can save data collection time and increase the efficiency of research by quickly obtaining the various data required for research. This section will introduce some of the Security datasets that are accessible on the Internet and facilitate section IV of the research results based on a more comprehensive understanding.

4.4.1 DARPA INTRUSION DETECTION DATA SETS

- DARPA Intrusion Detection Data Sets, which are under the direction of DARPA and AFRL/SNHS, are collected and published by The Cyber Systems and Technology Group (formerly the DARPA Intrusion Detection Evaluation Group) of MIT Lincoln Laboratory for evaluating computer network intrusion detection systems.
- The first standard dataset provides a large amount of background traffic data and attack data. It can be downloaded directly from the website. Currently, the dataset primarily includes the following three data subsets:
- 1998 DARPA Intrusion Detection Assessment Dataset: Includes 7 weeks of training data and 2 weeks of test data.
- 1999 DARPA Intrusion Detection Assessment Dataset: Includes 3 weeks of training data and 2 weeks of test data.
- 2000 DARPA Intrusion Detection Scenario-Specific Dataset: Includes LLDOS 1.0 Attack Scenario Data, LLDOS 2.0.2 Attack scenario data, Windows NT attack data.

4.4.2 KDD CUP 99 DATASET

The KDD Cup 99 dataset is one of the most widely used training sets; it is based on the DARPA 1998 dataset. This dataset contains 4 900 000 replicated attacks on record.

No	Features	Types	No	Features	Types
1	Duration	Continuous	22	Is_guest_login	Symbolic
2	Protocol type	Symbolic	23	Count	Continuous
3	Service	Symbolic	24	Srv_count	Continuous
4	Flag	Symbolic	25	Serror_rate	Continuous
5	Src_bytes	Continuous	26	Srv_serror_rate	Continuous
6	dst_bytes	Continuous	27	Rerror_rate	Continuous
7	Land	Symbolic	28	Srv_serror_rate	Continuous
8	Wrong_fragment	Continuous	29	Same_srv_rate	Continuous

9	Urgent	Continuous	30	diff_srv_rate	Continuous
10	Hot	Continuous	31	Drv_diff_host_rate	Continuous
11	Num_failed_logins	Continuous	32	Dst_host_count	Continuous
12	Logged in	Symbolic	33	Dst_host_srv_count	Continuous
13	Num_compromised	Continuous	34	Dst_host_same_srv_count	Continuous
14	Root_shell	Continuous	35	Dst_host_diff_srv_count	Continuous
15	Su_attempted	Continuous	36	Dst_host_same_srv_rate	Continuous
16	Num_root	Continuous	37	Dst_host_diff_host_rate	Continuous
17	Num_file_creations	Continuous	38	Dst_host_serror_rate	Continuous
18	Num_shells	Continuous	39	Dst_host_srv_serror_rate	Continuous
19	Num_access_files	Continuous	40	Dst_host_rerror_rate	Continuous
20	Num_outbond_cmds	Continuous	41	Dst_host_srv_rerror_rate	Continuous

There is one type of the normal type with the identity of normal and 22 attack types, which are divided into five major categories: DoS (Denial of Service attacks), R2L (Root to Local attacks), U2R (User to Root attack), Probe (Probing attacks) and Normal. For each record, the KDD Cup 99 training dataset contains 41 fixed feature attributes and a class identifier. Of the 41 fixed feature attributes, seven characteristic properties are the symbolic type; the others are continuous In addition, the features include basic features (No.1No.10), content features (No.11No.22), and traffic features (No.23No.41) as shown in Table 2. The testing set has specific attack types that disappear in the training set, which allows it to provide a more realistic theoretical basis for intrusion detection.

To date, the KDD Cup '99 dataset remains the most thoroughly observed and freely available dataset, with fully labeled connection records spanning several weeks of network traffic and a large number of different attacks Each connection record contains 41 input features grouped into basic features and higher-level features. The basic features are directly extracted or derived from the header information of IP packets and TCP/UDP segments in the tcp dump files of each session. The listless for tcp dump from the DARPA training data were used to label the connection records. The so-called content-based higher-level features use domain knowledge to look specifically for attacks in the actual data of the segments recorded in the tcp dump less. These address `r2l' and `u2r' attacks, which occasionally either require only a single connection or are without any prominent sequential patterns. Typical features include the number of failed login attempts and whether root access was obtained during the session. Furthermore, there are time-based and connection-based derived features to address `DOS' and `probe' attacks. Time based features examine connections within a time window of two seconds and provide statistics about these. To provide statistical information about attacks exceeding a two-second time-window, such as slow probing attacks, connection-based features use a connection window of 100 connections. Both are further split into same-host features, which provide statistics about connections with the same destination host, and same-service features, which examine only connections with the same service.

The KDD Cup '99 competitions provides the training and testing datasets in a full set and also provides a so-called `10%' subset version. The `10%' subset was created due to the huge amount of connection records present in the full set; some `DOS' attacks have millions of records. Therefore, not all of these connection records were selected. Furthermore, only connections within a time-window of five minutes before and after the entire duration of an attack were added into the `10%' datasets. To achieve approximately the same distribution of intrusions and normal traffic as the original DARPA dataset, a selected set of sequences with ‘nor-mal' connections were also left in the `10%' dataset. Training and test sets have different probability distributions. The full training dataset contains nearly five million records.

The full training dataset and the corresponding`10%' both contain 22 different attack types in the order that they were used in the 1998 DARPA experiments. The full test set, with nearly three million records, is only available unlabeled; however, a `10%' subset is provided both as unlabeled and labeled test data. It is specified as the `corrected' subset, with a different distribution and additional attacks not part of the training set. For the KDD Cup'99 competition, the `10%' subset was intended for training. The `corrected' subset can be used for performance testing; it has over 300,000 records containing 37 different attacks.

4.4.3 C- NSL-KDD DATASET

The NSL-KDD dataset is a new version of the KDD Cup 99 dataset. The NSL-KDD dataset improves some of the limitations of the KDD Cup 99 dataset. The KDD 1999 Cup Dataset Intrusion Detection Dataset was applied to the 3rd International Knowledge Discovery and Data Mining Tools Contest. This model identifies features between intrusive and normal connections for building network intrusion detectors. In the NSL-KDD dataset, each instance has the characteristics of a type of network data. It contains 22 different attack types grouped into 4 major attack types, as shown in Table 3. The dataset covers the KDD Train C dataset as the training set and KDD Test C and KDDTest21 datasets as the testing set, which has different normal records and four different types of attack records, as shown in Table 4. TheKDDTest21 dataset is a subset of the KDD Test C and is more difficult to classify.

4.4.4 ADFA DATASET

The ADFA data set is a set of data sets of host level intrusion detection system issued by the Australian de-fence academy (ADFA), which is widely used in the testing of intrusion detection products. In the dataset, various system calls have been characterized and marked for the type of attack.

The data set includes two OS platforms, Linux (ADFA-LD) and Windows (ADFA-WD), which record the order of system calls. In the case of ADFA-LD, it records the invocation of operating system for a period of time. such as the application of system resources, operating equipment, speaking, reading and writing, to create a new process, etc. User space requests, kernel space is responsible for execution, and these interfaces are the bridge between user space and kernel space. ADFA-LD is marked for the attack type, as shown in the figure. Linux system, user space by making system calls to kernel space to produce soft interrupts, so that the program into the kernel state, perform corresponding operations. There is a corresponding system call number for each system call. It contains 5 different attack types and 2 normal types.

Type of attacks in NSL-KDD

Types of Attack	Attacks in NSL-KDD Training Set
Dos	Back, Neptune, smurf, teardrop, land, pod
Probe	Satan, portsweep, ipsweep, nmap
R2L	Waremaster,warezelient,ftpwrite,guesspassword,imap,multihop,phf,spy
U2R	Rootkit, butter ,overflow, load-module, Perl.

NSL-KDD

	Total	Normal	Dos	Probe	R2L	U2L
KDD Train	125973	67343	45927	11656	995	52
KDD Test	22544	9711	7458	2421	2754	200
KDD $\boldsymbol{Test^{-21}}$	11850	2152	4342	2402	2754	200

Type of attacks in ADFA-LD.

Attack Type	Data size	Note type
Training	833	Normal
Validation	4373	Normal
Hydra-FTP	162	Attack
Hydra-SSh	148	Attack
Add user	91	Attack
Java-Meter perter	75	Attack

ALGORITHAM

CHAPTER-5

5.5 ML AND DL ALGORITHM FOR CYBERSECURITY

This section is divided into two parts. The first part introduces the application of traditional machine-learning algorithms in network security. The second part introduces the application of deep learning in the field of cyber security. It not only describes the research results but also compares similar studies.

5.5.1 MODIFIED RANDOM FOREST

Modified Random Forest (MRF) is one of the most robust and accurate methods in all machine learning algorithms. It primarily includes Support Vector Classification (SVC) and Support Vector Regression (SVR). The SVC is based on the concept of decision boundaries. A decision boundary separates set of instances having different class values between two groups. The SVC supports both binary and multi-class classifications. The support vector is the closest point to the separation hyper plane, which determines the optimal separation hyper plane. In the classification process, the mapping input vectors located on the separation hyper plane side of the feature space fall into one class, and the positions fall into the other class on the other side of the plane. In the case of data points that are not linearly separable, the MRF uses appropriate kernel functions to map them into higher dimensional spaces so that they become separable in those spaces Kotpalliwar and Wangi choose two representative datasets ``Mixed'' and ``10% KDD Cup 99'' datasets. The RBF is used as a kernel function for MRF to classify DoS, Probe, U2R, and R2L datasets. The study calculates parameter values related to intrusion-detector performance evaluation. The validation accuracy of the ``mixed'' dataset and the classification accuracy of the ``10% KDD'' dataset were estimated to be 89.85% and 99.9%, respectively. Unfortunately, the study did not assess accuracy or recall except for accuracy.

Saxena and Richariya proposed a Hybrid PSO-MRF approach for building IDS. The study used two feature reduction techniques: Information Gain and BPSO. The 41 attributes reduced to 18 attributes. The classification performance was reported as 99.4% on the DoS, 99.3% on Probe or Scan, 98.7% on R2L, and 98.5% on the U2R. The method provides a good detection rate in the case of a Denial of Service (DoS) attack and achieves a good detection rate in the case of U2R and

R2L attacks. However, the precision of Probe, U2R and R2L is 84.2%, 25.0% and 89.4%, respectively. IN other words, the method provided by the essay leads to a higher false alarm rate.

Pervez and Farid [30] propose a filtering algorithm based on a Modified Random Forest (MRF) classifier to select multiple intrusion classification tasks on the NSL-KDD intrusion detection dataset. The method achieves 91% classification accuracy using only three input features and 99% classification accuracy using 36 input features, whereas all 41 input features achieve 99% classification accuracy. The method performed well on the training set with an F1-score of 0.99. However, in the test set, the performance is worse; the F1- score is only 0.77. With poor generalization, it cannot effectively detect unknown network intrusions.

With the help of the fuzzy C-means clustering technique, the heterogeneous training data are collected into homogeneous subsets, reducing the complexity of each subset, which helps to improve detection accuracy. After the initial clustering, ANNs are trained on the corresponding homogeneous subsets and use the linear MRF classifier to perform the final classification. The experimental results obtained with the calibrated KDD CUP 1999 dataset show the effectiveness of this method. In the same work, the KDD Cup 99 dataset is divided into 4 subsets according to different intrusion types and trained separately; DoS and PROBE attacks have a higher frequency and can be effortlessly separated from normal activity. In contrast, U2R and R2L attacks are embedded in the data portion of the packet, making it difficult to achieve detection accuracy on both types of attacks. The technique has attained a consistent peak score for all types of intrusions. Overall accuracy of the DoS, Probe, R2L and U2R categories was 99.66%,98.55%, 98.99% and 98.81%, respectively. Compared with other reported intrusion detection approaches, this method is better in classification effect, but the trained classifier cannot effectively detect the abnormal in the actual network.

Yan and Liu attempt to use a direct support vector classifier to create a transudative method and introduce the simulated annealing method to degenerate the optimization model. The study used a subset of the DARPA 1998 dataset. For DoS-type attacks, 200 normal samples and 200 attack samples are selected; the feature dimension is 18, and samples are randomly divided into a training set and a test set according to the ratio 6:4. The experimental results show that the accuracy, FPR and precision are 80.1%, 0.47% and 81.2%, respectively. The dataset used in this study is too small, and the classification results are not very satisfactory.

Using the same dataset, Kokila et al. focus on DoS attacks on the SDN controller. A variety of machine learning algorithms were compared and analyzed. The MRF classifier has a lower false alarm rate and a higher classification accuracy of 0.8% and 95.11%, respectively. On the basis of a short sequence model, Xia et al. applied a class MRF algorithm to ADFA-LD. Due to the short sequence removes duplicate entries, and between the normal and abnormal performed better separability, so the technology can reduce the cost of computing at the same time to achieve an acceptable performance limit, but individual type of attack mode recognition rate is low.

5.5.2 K-NEARESTNEIGHBOR

The KNN classifier is based on a distance function that measures the difference or similarity between two instances. The standard Euclidean distance d (x, y) between two instances x and y is defined as:

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

Where, x^k is the k^th featured element of instance x, y_k is thef^t featured element of the instance y and n is the total number of features in the dataset.

Assume that the design set for KNN classifier is the total number of samples in the design set is S. Let C D {C1; C2: CL} are the L distinct class labels that are available in S. Let x be an input vector for which the class label must be predicted. Let$y_k$denote the k^th vector in the design set S. The KNN algorithm is to find the k closest vectors in design set S to input vector x. Then the input vector x is classified to class c_j if the majority of the k closest vectors have their class as c_j.

Rao and Swathi [36] used Indexed Partial Distance Search k-Nearest Neighbor (IKPDS) to experiment with various attack types and different k values (i.e., 3, 5, and 10). They randomly selected 12,597 samples from the NSL-KDD dataset to test the classification results, resulting in

99.6% accuracy and faster classification time. Experimental results show that IKPDS, and in a short time **Network Intrusion Detection Systems** (NIDS), have better classification results. However, the study of the test indicators of the experiment is not perfect; it did not consider the precision and recall rate.

Sharie al. presents the K-Means and KNN combination intrusion detection system. First, the input invasion data (NSL-KDD) are preprocessed by principal component analysis to select the best 10 important features. Then, these preprocessed data are divided into three parts and fed into the k-means algorithm to obtain the clustering centers and labels. This process is completed 20 times to select the best clustering scheme. These cluster centers and labels are then used to classify the input KDD data using simple KNNs. In the experiment, two methods were used to compare the proposed method and the results of KNN. These measures are based on the accuracy of the true detection of attack and attack type or normal mode. Implement two programs to investigate the results. In the first case, the test data are separated from the train data, whereas in the second case, some test data are not substituted from the training data. However, in any case, the average accuracy of the experiment is only approximately90%, and it did not consider the precision and recall rate.

Saponified and Ahmadinejad studied some new techniques to improve the classification performance of KNN in intrusion detection and evaluate their performance on NSL-KDD datasets. The farthest neighbor (k-FN) and nearest neighbor (KNN) are introduced to classify the data. When the farthest neighbor and the nearest neighbor have the same category label, the second nearest neighbor of the data is used for discrimination. Experimental results show that this method has been improved in terms of accuracy, detection rate and reduction of failure alarm rate. Because the experimental results in this paper only provide a histogram, the accuracy of this method can only be roughly estimated. The detection rate and false alarm rate are 99%, 98% and 4%, respectively. However, the study did not identify specific types of attacks in abnormal situations.

To reduce the false alarm rate, developed a knowledge-based alarm verification method, designed an intelligent alarm filter based on multi-level k-nearest neighbor classifier and filtered out unwanted alarms. Expert knowledge is a key factor in evaluating alerts and deter mining rating thresholds. Alert filters classify incoming alerts into appropriate clusters for tagging through expert knowledge rating mechanisms. Experts will further analyze the effect of different classifier settings on classification accuracy in the evaluation of the performance of alarm filters in real datasets and network environments, respectively. Experimental results show that when K=5, the alarm filters can effectively filter out multiple NID Salaams.

In the same work, the study initially trained the filter using the DARPA 1999 dataset and evaluated it in a network environment built by Snort and Wire shark. Before Snort was deployed on the internal network, Snort was designed to detect various types of network attacks. Wire shark was implemented before Snort and was responsible for recording network packets and providing statistics. KNN based smart alarm filters are deployed behind Snort to filter Snort alarms. Real-world web traffic will pass through Wire shark and reach Snort. A snort checks network packets and generates alerts. Thereafter, all generated Snort alarms will be forwarded to intelligent KNN based alarm filters for alarm filtering. Experiments use the accuracy and F-score as indicators; the averages of results were 85.2% and 0.824, respectively.

Vishwakarma et al. propose a KNN intrusion detection method based on the ant colony optimization algorithm (ACO), pre training the KDD Cup 99 dataset using ACO, and studies on the performance of KNN-ACO, BP Neural network and Modified Random Forest for comparative analysis with common performance measurement parameters (accuracy and false alarm rate). The overall accuracy reported for the proposed method is 94.17%, and the overall FAR is 5.82%. Unfortunately, the dataset used for this study was small, with only 26,167 samples participating in the training.

Another study used KNN for intrusion detection on the same KDD Cup 99 dataset in an approach similar to that of Vishwakarma. The main difference is that the k-NN, MRF, and pd APSO algorithms are mixed to detect intrusions. The experimental results show that mixing different classifiers can improve classification accuracy. The statistical results show that the classification accuracy is 98.55%. Other than accuracy, the study did not count other indicators.

5.5.3 DECISION TREE

A decision tree is a tree structure in which each internal node represents a test on one property and each branch represents test output, with each leaf node representing a category. In machine learning, the decision tree is a predictive model; it represents a mapping between object attributes and object values. Each node in the tree represents an object, each divergence path represents a possible attribute value, and each leaf node corresponds to the value of the object represented by the path from the root node to the leaf node. The decision tree only has a single output; if want complex output, can establish an independent decision tree to handle different outputs. Commonly used decision tree models are ID3, C4.5 and CART.

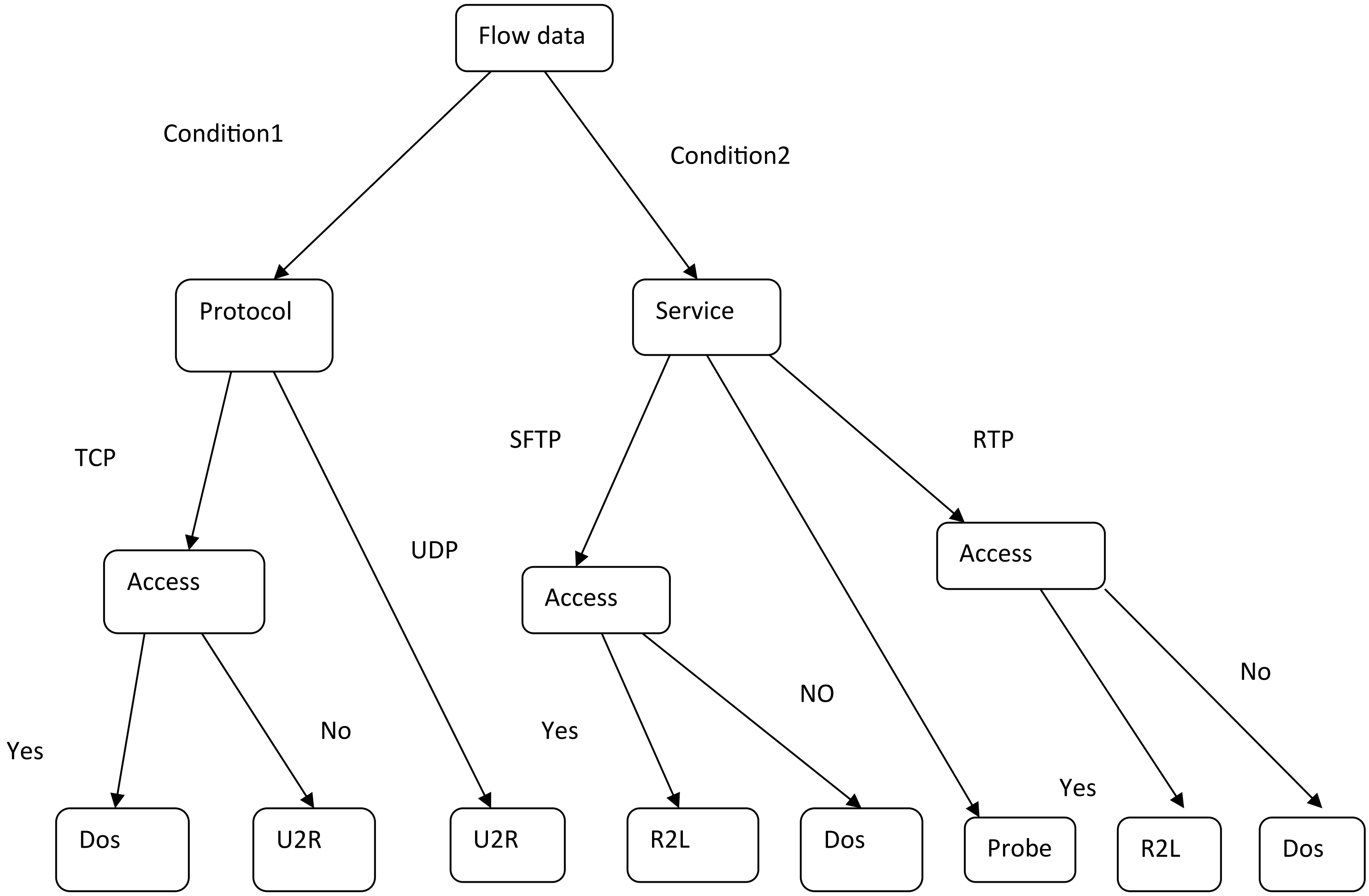
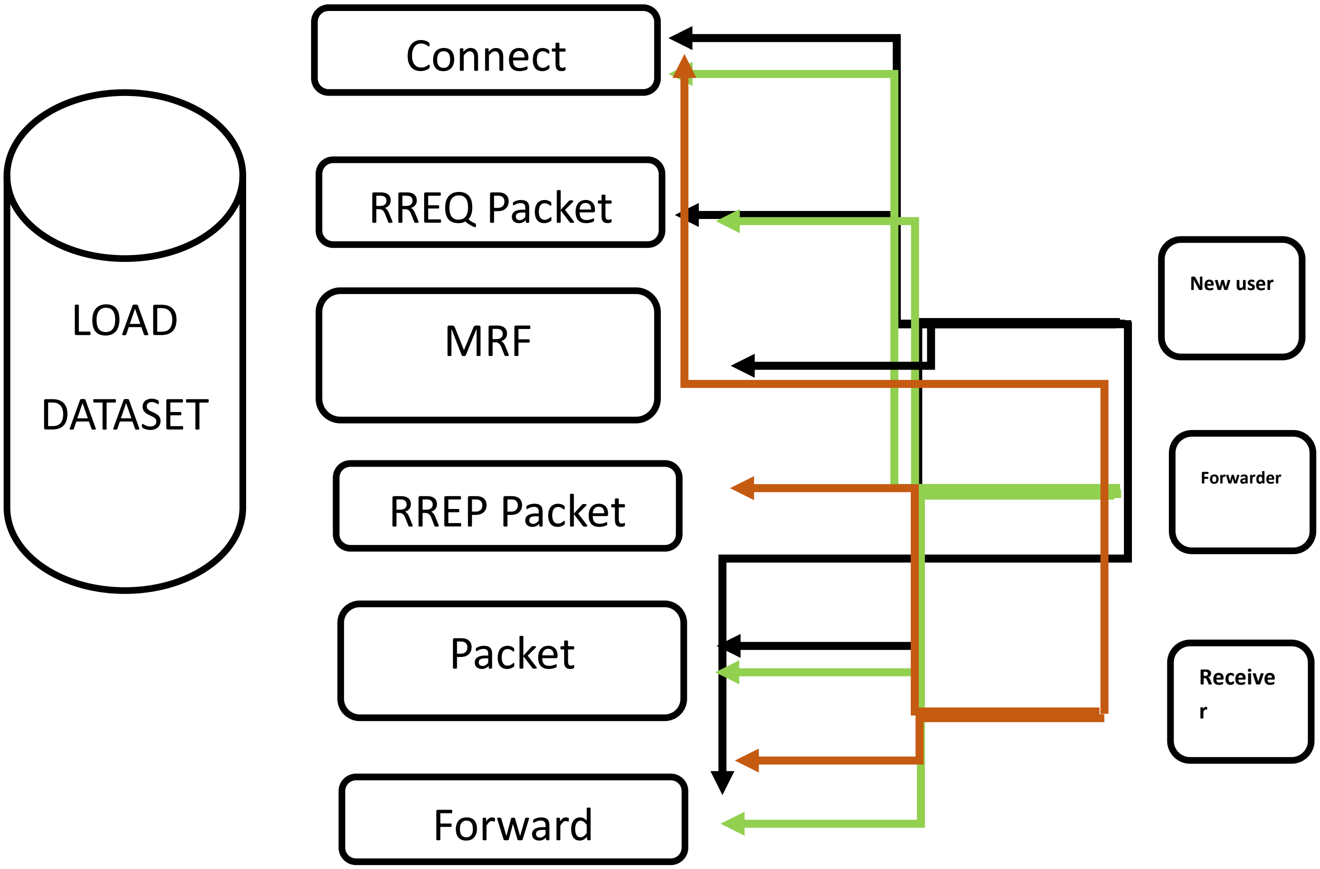


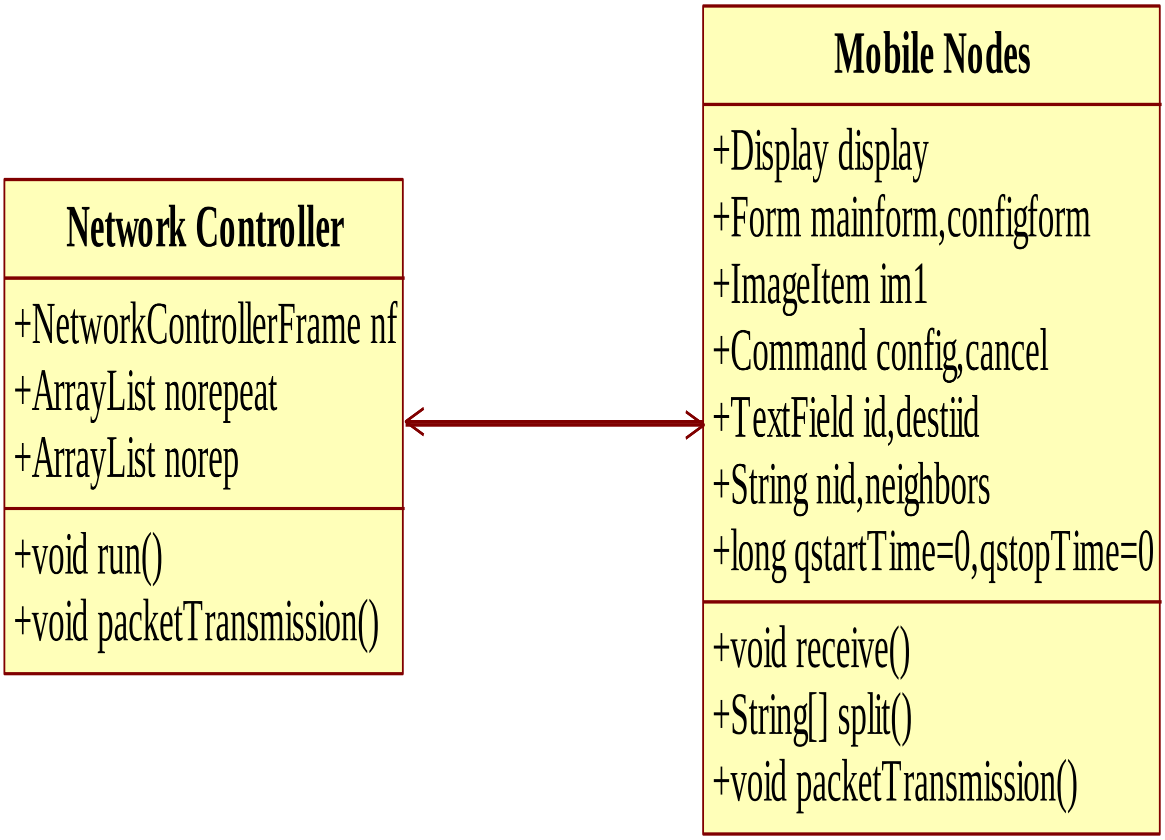
Figure.1 An example decision tree

As shown in Fig.1, the decision tree classifies the samples through the conditions of training, and has better detection accuracy for known intrusion methods, but is not suitable for detection of unknown intrusion.

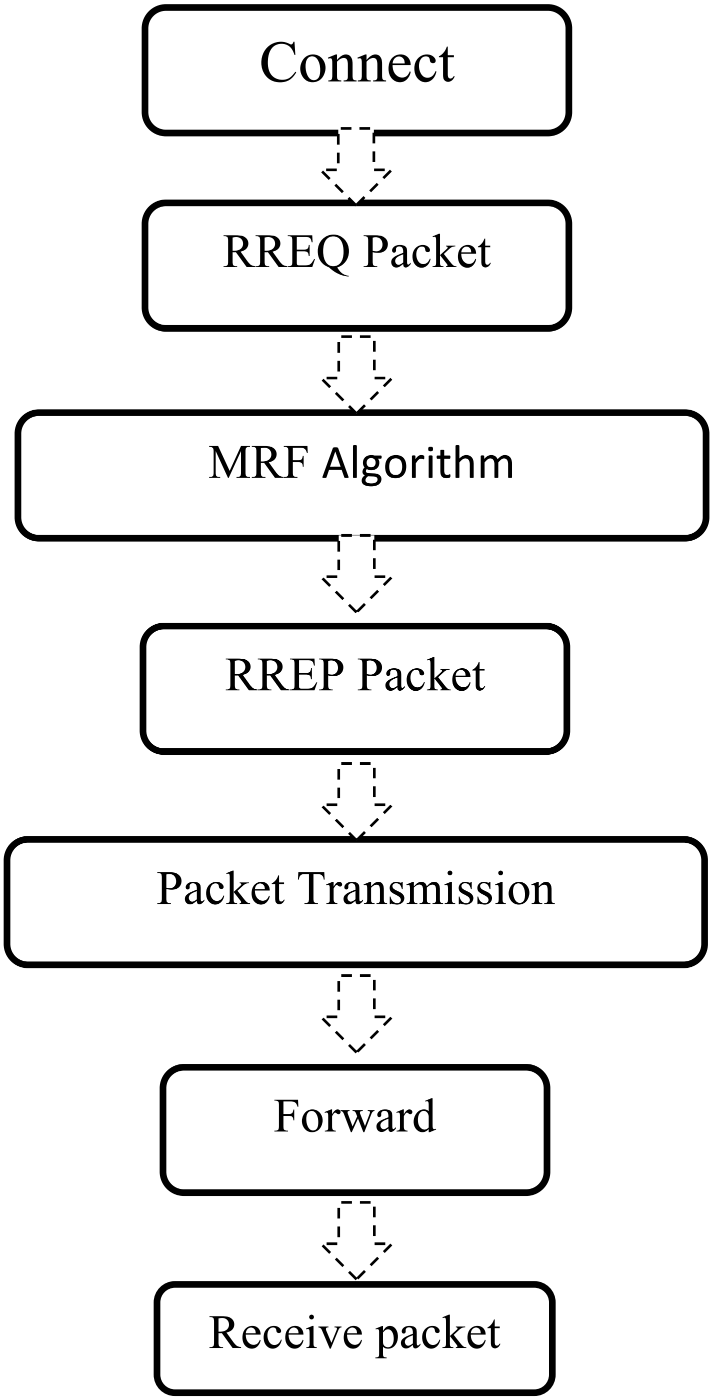
USECASE DIAGRAM:



CLASS DIAGRAM:



ACTIVITY DIAGRAM:



AEXPERIMENTAL SETUP

CHAPTER-6

6.1EXPERIMENTAL SETUP

This part engages in a simulation to evaluate the future algorithm. The research has been conducted on the platform of individual computer with 1.5 GHz CPU and 4GB RAM. The operating system is Windows 7, and simulation programs are executed in Java with Mat lab 2014. The research examines a large number of academic intrusion detection studies based on machine learning and deep learnings as shown in Table 5. In these studies, many imbalances appear and expose some of the problems in this area of research, largely in the following areas:

- The benchmark datasets are few, although the same dataset is used, and the methods of sample extraction used by each institute vary.
- The evaluation metrics are not uniform, many studies only assess the accuracy of the test, and the result is one-sided. However, studies using multi criteria evaluation
- Less consideration is given to deployment efficiency, and most of the research stays in the lab irrespective of the time complexity of the algorithm and the efficiency of detection in the actual network.

In addition to the problem, trends in intrusion detection are also reflected.

- The study of hybrid models has been becoming hot in recent years, and better data metrics are obtained by reasonably combining different algorithms.
- The advent of deep learning has made end-to-end learning possible, including handling large amounts of data without human involvement. However, the ne-tuning requires many trials and experience; interpretability is poor.
- Papers comparing the performance of different algorithm saver time are increasing year by year, and increasing are beginning to value the practical significance of algorithms and models.
- A number of new datasets are in the school's charge, enriching the existing research on cyber security issues, and the best of them is likely to be the benchmark dataset in this area. The problems and trends described above also provide a future for intrusion detection research

CONCLUSION

CHAPTER-7

7.1 CONCLUSION

In this project, we have proposed a new approach to detect the emergence of topics in a social network stream. The basic idea of our approach is to focus on the social aspect of the Grids reflected in the mentioning behavior of users instead of the textual contents. We have combined the proposed mention model with the MRF change-point detection algorithm. The Micro Grid Anomaly Detection proposed is an approximation to the NML code length that can be computed in a sequential manner. The MRF proposed further employs discounting in the learning of the AR models. As a final step in our method, we need to convert the change-point scores into binary alarms by thresholding. Since the distribution of change-point scores may change over time, we need to dynamically adjust the threshold to analyze a sequence over a long period of time. In this subsection, we describe how to dynamically optimize the threshold using the method of dynamic threshold optimization proposed. In DTO, we use a one-dimensional histogram for the representation of the score distribution. We learn it in a sequential and discounting way.

.

BIBLIOGRAPHY

CHAPTER -8

8.1 REFERENCES

1. Sharafaldin, I, Lashkari,A.H and Ghorbani, A.A, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, (2022).
2. Gharib, A., Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A., “An Evaluation Framework for Intrusion Detection Dataset”. 2016 IEEE International Conference Information Science and Security (ICISS), pp. 1-6, (2021)
3. Gil, G.D., Lashkari, A.H., Mamun, M. and Ghorbani, A.A., “Characterization of encrypted and VPN traffic using time-related features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy, pp. 407-414, (2021).
4. Moustafa, N. and Slay, J., “The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 dataset”. Information Security Journal: A Global Perspective, 25(1-3), pp.18-31, (2022).
5. Moustafa, N. and Slay, J., “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). IEEE Military Communications and Information Systems Conference (MilCIS), pp. 1-6, (2021).
6. Pongle, Pavan, and GurunathChavan. "A survey: Attacks on RPL and 6LoWPAN in IoT." IEEE International Conference on Pervasive Computing, (2022).
7. Oh, Doohwan, Deokho Kim, and Won Woo R, "A malicious pattern detection engine for embedded security systems in the Internet of Things." Sensors, pp, 24188-24211, (2021).
8. Mangrulkar, N.S., Patil, A.R.B. and Pande, A.S., “Network Attacks and Their Detection Mechanisms: A Review”. International Journal of Computer Applications, 90(9), (2021).
9. Kasinathan, P., Pastrone, C., Spirito, M. A., &Vinkovits, M. “Denialof-Service detection in 6LoWPAN based Internet of Things.” In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 600-607, (2021).

APPENDIX

CHAPTER -9

9.1 SCREEN LAYOUT

```
==================================

Imbalanced Data with 10000 records

==================================

         Duration    Protocol_type  ...    Dst_host_srv_rerror_rate
Class

0             0          tcp  ...             0.0    back

1             0          tcp  ...             0.0    back

2             0          tcp  ...             0.0    back

3             0          tcp  ...             0.0    back

4             0          tcp  ...             0.0    back

...          ...          ...  ...             ...     ...

9995          0          tcp  ...             0.0  normal

9996          0          tcp  ...             0.0  normal

9997          0          tcp  ...             0.0  normal

9998          0          tcp  ...             0.0  normal

9999          0          tcp  ...             0.0  normal
```

9.2 REPORTS

```
=============================================================================

PCA-RF Predicted class label values for Testing Dataset

=============================================================================

 [ 9 9 11 11 20  9  9 11 11 11  9 11 11  9  9 11 11  9  9 11 11 11  0  9

 11 11 11  9 18 11 11 11  9 11  9 18  1 11 11  9 11  0 11 15 20 11 11 11

 11 11 11 11 20 11 11 11 9 20 18 11 18 9 11 11 21 11  9  9 21  9  0 11

11  9 11  9  5  9  5  9 11 11 11  9 11  5 11 20 11  9 11 11 11 20 11 11

  9 11 11 9  9 11 11 11 11 11 14 15 11 11 11 11 11  0 21  9 11  9 11 11

 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..

 21 11 11 11 11 11 20  9  0 11 11 11 11  5 11 11  9 11  0  9  0  5 20 11

  0 11 18 11 11 11 11  9 11  9  9  9 18  9 20 11]

PCA-RF Accuracy:  98.11075557935591  %

SVM Accuracy:  92.87808766147013  %
```