PERFORMANCE ANALYSIS OF DEEP LEARNING ALGORITHMS FOR INTRUSION DETECTION SYSTEM(IDS) OF IoT SECURITY

by

Yeamin Jahan Fiha
17101475
Mithila Farjana
17201042
Syed Rifat Bin Masum
17201063
Mehedi Bin Sarwar Soron
17201064
Mia Fahim Hasan Alif
17301206

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

We would like to declare that we have submitted this thesis paper to BRAC University in order to complete our Bachelor's degree.

- This paper was not submitted to any other journal or conference.
- This is the first time we've submitted a paper like this.
- Our group members are responsible for every aspect of completing this paper.
- We attempted to cite all of the sources that we used in this project.

Student's Full Name & Signature:

Yeamin

Yeamin Jahan Fiha

17101475

Mithila Farjana

Mithila Farjana

17201042

Rifat

Syed Rifat Bin Masum

17201063

Soron

Mehedi Bin Sarwar Soron

17201064

Fahim

Mia Fahim Hasan Alif

17301206

Approval

”Performance Analysis of Deep Learning Algorithms For Intrusion Detection System(IDS) of IoT Security” this thesis title is submitted by

1. Yeamin Jahan Fiha (17101475)
2. Mithila Farjana (17201042)
3. Syed Rifat Bin Masum (17201063)
4. Mehedi Bin Sarwar Soron (17201064)
5. Mia Fahim Hasan Alif (17301206)

Of Fall’2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 16, 2022.

Examining Committee:

Supervisor:
(Member)

[illegible]

Muhammad Iqbal Hossain, PhD

Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

[illegible]

Mr. Faisal Bin Ashraf

Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

[illegible]

Md. Golam Rabiul Alam, PhD

Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Professor and Chairperson (CSE)
Department of Computer Science and Engineering
Brac University

Ethics Statement

Our personal explorations have led us to the conclusion that this assertion is based on fact. In this paper, all of the documents and archives that have been used are properly cited and referenced. This work has never before been submitted to another college or academic institution, in whole or in part, for the purpose of receiving a degree.

Abstract

The Internet of Things (IoT) is a term that refers to billions of actual gadgets all over the universe that are connected to the network, participating in all social events, and sharing data. It is the most recent technological advancement in history. In addition, Internet of Things devices have sensors and small PC processors that generate applications based on the data gathered by the sensors through Artificial Intelligence (AI). Fundamentally, Internet of Things (IoT) devices are more modest than anticipated PCs, are connected to the internet, and are vulnerable to viruses and hacking. Similarly, there are legitimate concerns about risks associated with the expansion of the Internet of Things, notably in the areas of safety and security. Nowadays, the intrusion detection system for Internet of Things devices is a critical concern. Entering the computer environment is an extremely dangerous and unpredictable activity that has existed since the invention of computer technology. Many security measures have been implemented over the past three decades, but as technology has progressed, so have the threats to national security. Because the world is increasingly reliant on computers, whether directly or indirectly, it is critical to prevent potentially dangerous activities and attacks that could jeopardize computer infrastructure. IDS and IPS are two often used security solutions to protect computer resources, particularly those on a network. To create a secure Internet of Things deployment, a variety of security principles need to be followed at each tier. In this case, the impact of artificial intelligence on the internet security of Internet of Things devices is likely to alter the traditional risk assessments. Aside from that, the field of Artificial Intelligence (AI) is rapidly expanding, propelled by shifts in neuron organization and deep learning. In addition to this, we have employed Deep Learning Approach to detect the intrusion of IoT gadgets as well as other types of intrusion. Deep Learning is a sub-sector of artificial intelligence and machine learning (ML) that mimics the way the human brain works in terms of data processing and making effective decisions. As a result, it is playing an important role in the detection of intrusion from devices that are linked to the internet. The IDS (Intrusion Detection System) of Internet of Things System utilizing Deep Learning techniques is the subject of our thesis. Our article provides a complete analysis of security policies, technical obstacles, and solutions for Internet of Things (IoT) security protection. In addition, we have employed two datasets, namely KDD-99 and NSL-KDD, as well as three methods, including RNN, LSTM, and GRU, in our paper. The proposed model was tested and evaluated as a consequence, and the results demonstrate that the model is extremely accurate when it comes to distinguishing interruptions in Internet of Things devices..

Keywords: Deep learning, Intrusion Detection System, IoT, Security, RNN, GRU, LSTM, KDD-99, NSL-KDD, LSTM, GRU, IDS, Vanilla LSTM, Bidirectional LSTM ,Stacked LSTM

Dedication

To our respected faculty members who have taught us over the past four years, especially our honorable supervisor Muhammad Iqbal Hossain (PhD) sir, we would like to dedicate this thesis.

Acknowledgement

Let us begin by thanking Allah, the Almighty, for providing us with this opportunity to finish our thesis on time. Our deepest gratitude goes out to Dr. Muhammad Iqbal Hossain sir, our respected supervisor, for his untiring assistance, guidance, and encouragement throughout the process. Moreover special thanks to Syed Zamil Hasan Shoumo Sir, a revered member of our department, was important in helping us to complete this thesis. To overcome any difficulties, they gave enthusiasm and instruction. We could call on them at any time if we needed support. We also like to thank our BRAC University for giving us the opportunity and assistance we needed to finish this thesis.

Table of Contents

List of Figures

List of Tables

Nomenclature

The following rundown depicts a few images and condensing that will be subsequently utilized inside the body of the report

IoT	Internet of Things
MLP	Multilayer perceptron
RNN	Recurrent neural network
LSTM	Long Short-Term Memory
GRU	Gated recurrent Unit
NSL-KDD	Network Security Laboratory- Data Mining & Knowledge Discovery
KDD	Knowledge Discovery and Data Mining
ANN	Artificial Neural Network
DOS	Denial of Service
IDS	Intrusion Detection System
AI	Artificial intelligence
IPS	Internet Protocol Security
ML	Machine Learning
DBN	Deep belief network
DNN	Deep neural network
DDoS	Distributed-Denial-of-Service
SIDS	Signature-placed IDS
IDPs	Intrusion Detection and Prevention System
GDP	Gross Domestic Product

Chapter 1

Introduction

Improvement of the Internet of things is conceivably dependent on the union in advance, a specific draw established by the past time in registering, concurrence, and an application plan. Because of this, the Internet of Things' ensuing circle of influence has swiftly expanded to encompass the whole human being race. Among the Internet of Things gadgets employed to assist us with our daily activities are advanced cells and home associates such as Google Play, brilliant automobiles, building computerization frameworks, electric vehicles such as robots, environmental monitoring, and entertainment.[31]
The rapid growth of technological improvements has made life easier while exposing a slew of safety concerns. With the development of the internet over the years, the number of cyberattacks has increased in number as well. The Interruption Detection System (IDS) is one of the most important layers to consider when it comes to data security. Businesses benefit from IDS because it promotes a healthy environment and prevents dubious organizational exercises. In this research, we attempt to apply a variety of algorithms and datasets to determine which one produces the most relevant result for intrusion detection applications.[11]
According to a Gartner event evaluation, approximately 25 billion connected things will be active continuously in 2020. These related gadgets help make regular activities more enjoyable and create beautiful arrangements. When developing solutions for the Internet of Things, one must consider the different associated devices, complexities, competing patterns, and types that must be considered. The current security conventions are only applicable for powerful PCs that are used for brief meetings or presentations. It is not practical to employ a similar insurance method in meetings that go for an extended period of time. Thus, Internet of Things (IoT) devices become enticing targets for programmers, putting our lives in danger by posing unexpected risks to them.[8]
These difficulties in IoT can be dealt with by employing the concepts of ”falling” and ”gathering” to develop innovative and adequate security measures. Massive passed-on structures have shown their inconsistencies, and ”Flexible Lightweight” plans have proven their worth in dealing with them. However, it is nearly impossible to answer for this type of device in an organization because there are so many of them. In any event, a sensible approach would be to gather data from the IoT devices themselves. In order to cultivate flexible reactions, this system can be supported by human-created mental abilities. Utilizing the enormous amounts of data generated by IoT devices, AI and data analytics technologies train employees to

help boost productivity at their organizations. IoT networks can use IoT networks to identify and prevent potentially catastrophic events and peculiar behaviors by recognizing design flaws, looking for inconsistencies, and conducting sociological investigations.[8] One aspect of a practical security check approach is the Mark-based and Inconsistency-based Intrusion Detection Framework. Few resource owners have IDS/IPS sent and built appropriately between enterprise IT and ICS businesses.[12] The detection module evaluates and analyses the formatted data obtained from the data collection model to detect intrusion attempts and then transmits the events labeled as malicious to the reaction module. An anomaly-passionate disclosure, specification-passionate disclosure, and misuse-passionate disclosure are three different sorts of intrusion disclosure methodologies. Anti-intrusion techniques based on anomalies identify the system's normal actions.[1] IDS has a slew of potential deep learning approaches. Since each strategy is given its own set of plan considerations, it may be hampered in its capacity to achieve outstanding execution in terms of viability and proficiency.[15] Using this theory, we have devised a new assessment that comprises a Many Layer strategy for our proposed notion, In this case, we attempted to use important knowledge estimations. An IoT network determines if a row in a system is "normal" or "offensive" based on its association information. The KDD99, NSL-KDD dataset, and various algorithms such as MLP, LSTM, and GRU are used in a varied number of covered-up hubs and learning speeds in AI research on-network data security.[8][17]

1.1 Motivation

Globally, the Internet of Things (IoT) is experiencing an inescapable update. Since the 2016 Dyn cyberattack, which was perpetrated by the erroneous Mirai Internet of Things botnet, IoT security has become a major source of concern for a large number of individuals. The danger posed by a large number of Internet-connected items has an impact on the security of the Internet of Things. However, it also threatens the whole Internet regular framework. Multiple rows of data are combined with the web layer in this example. The most generally known care risks examined by the IoT architecture include bonnets, DDoS intrusion, distant documentation, routing treatments, and information flow. Combatants attack Internet of Things devices, which are regarded as the first line of defense, but which are not sufficient because of the changeability and complexity of IoT systems.
At the moment, IDS has met the demand for its energy supplies. IDS is a notion that identifies systems that are being raided.There are no entry barriers to the Internet of Things (IoT). IDS's core consists of an agent, an investigative motor, and a reaction module. IDS are categorized into two main groups. First, there are two types of IDSs: one for the host and one for the internet. IDS for moderator-related activities include a ban on interruption conduct and review data. These kinds of Ids are generally used on large hosts to protect the host's security from all angles. Second, network-based IDS is more complicated, while the host-based IDS provides more precise data and has a lower probability of false alarms. Sadly, this reduces the application framework's efficiency and relies on log information.
Network-based IDS can detect unexpected behavior. As a result, neither the host arrangement nor the introduction of the corporate structure is altered in any way.

Whether or not the network-dependent association IDS hits the Mark, it will have little effect on everyday company operations. IDS is one of the problems. Signature-based IDSs (detect rapid relationship association section monitoring other organization fragments) are used. Encoded gatherings along the network are difficult to measure.
Security Information and Data Systems (SIDS) are systems that store a "signature" of previously known attacks. In order to compare these signatures with the ones in the data set, the signatures of the present exercises are deleted, and coordination techniques and convention conformance checks are utilized. They are assumed to be afflicted if they fit the criteria. In the absence of a net, framework events are investigated, and in the existence of one, the owners are checked, and alerts are brought up progressively.
A signature-placed IDS is unable to protect against the very first attacks since the name is not included in this IDS's data. Zero-day assaults, on the other hand, continue to rise. It follows that the total defensive system is losing effectiveness due to Signature Based IDS. Deviation occupied IDSs were progressed, unpopular IDSs became the norm, and this report was the most significant adjustment for this model's control due to this development.
In order to detect intruders, the deviation occupying IDS relies on artificial intelligence and engineering knowledge. An AI or ML tool is being developed that can take in information without the intervention of a human and discern between normal behavior in a system and unexpected or anomalous behavior. A machine can be prepared in a variety of methods, including through directed, solo, or supported learning. Naive Bayes classifier, artificial neural network, base vector engines, and linear regression are just a few examples of the tools that are employed in tool information. [4]. DBN, DNN, and recurrent RNN are the most common deep learning architectures.[13] Due to its high success rate, the Deep Learning technique is currently quite popular.

1.2 Problem Statement and Thesis Objectives

In this research, we compare the behavior of various deep learning algorithms on the KDD-99 cup and NSL KDDcup data sets. Additionally, we sought out the most effective intrusion detection techniques.

The research's primary aims are as follows:

- Having a working knowledge of the Internet of Things security system
- In order to comprehend the intrusion detection system
- In order to comprehend the deep learning algorithm
- In order to observe the behavior of different algorithms when applied to different datasets
- To observe the model results of various algorithms.
- To compare and contrast the various outcomes of various algorithms

1.3 Thesis Structure

Our thesis paper is divided into numerous sections. Background data, which includes a section on the literature and a description of the algorithm, is presented in Chapter 2 of this document. The information contained in literature reviews pertains to prior work and models. In addition, we have read a number of different research publications. In Chapter 3, we described our dataset and the methods we used to analyze it. The KDD999 dataset, as well as the NSL KDD dataset, were used in this study. We employed three types of pre-processing methods: the Train-Test split, the Feature Selection technique, and the Fold Cross approach. On our dataset in Chapter 4, we applied LSTM, GRU, and RNN models. First, we compared the performance of Stacked LSTM, Vanilla LSTM, and Bidirectional LSTM to see which one produced the best results. Then we combined the results of both LSTM and GRU to make a more accurate comparison. Towards the end of Chapter 5, we presented preliminary results of our analysis and compared our findings to determine which Deep Learning Model provided the best Intrusion Detection System.

Chapter 2

Background

2.1 Literature Review

With the rise of IoT devices and new attack methods, cyber security has become increasingly important in our day. To design an innovative real-world environment, security and privacy are essential. Because of insufficient security mechanisms, cyber-attacks are the most susceptible point of the Internet of Things. In the Internet of Things, these security protocols are dependent on network layers. The Internet of Things is plagued by multiple attacks because of the vast number of devices that are connected across these network levels. Each device generates data and distributes it to the rest of the world via the internet. For attacking IoT devices, hackers are always coming up with new and innovative tactics. For example, an intrusion detection system is used to identify these incursions of these IoT devices. However, an intrusion detection system is used to classify these attacks in network traffic using a deep or wide learning strategy in conjunction with a neural network algorithm.
IoT refers to the organization of specific devices that are networked via the internet. The Internet of Things, or IoT, refers to the gadgets that are connected to the internet. Sensors, programming, and other advancements are incorporated into the Internet of Things to allow devices to communicate and exchange information with one another, starting with one device and progressing to the next. This adaptable architecture is being developed step by step over the entire web environment. It is estimated that there are more than seven billion connected IoT devices in use today. Aside from that, it is expected by experts that this number will increase to more than 25 billion by 2030. As a result, Internet of Things (IoT) security is becoming important.[19]
The Industrial Internet of Things (IIoT) sector has made significant strides in recent years. Many industrial types of machinery are connected to the network as a result of the Internet of Things, and these machines store a wealth of important data. These Internet of Things systems become targets for hackers for a variety of reasons, including the data they contain. If hackers gain access to the Internet of Things system, they will be able to steal important information. They choose a method that is both simple and effective for detecting intrusion. They are attempting to detect intrusions using a deep learning approach. They have also discussed some of the difficulties associated with building Deep Learning-based IDS, such as the high cost, implementation difficulties, and lack of data. Training the neural network system can be difficult because the neural network system's success rate depends on

the amount of training it receives.[32]
Following the discussion in the paper below, various types of intrusion detection systems have been identified and discussed. After that, he went into detail about the significance of the Internet of Things. The Internet of Things is being used in a variety of fields, including education, health care, and industry. Because IoT devices are connected via a wireless network, it is possible for an intruder to gain access to the system and cause damage, and while the devices include security features such as encryption and decryption, these are insufficient for the overall security of the system. It is for this reason that intrusion detection is required. After that, the author discussed IDS and why it is crucial to have one in place. Suppose an intrusion occurs in the Internet of Things network. In that case, the IDS's responsibility is to identify the intrusion and notify the user of the intrusion before the intruder causes any damage to the system. Following that, the different types of IDS are discussed in detail throughout the study. In IoT systems, a variety of cyber threats might manifest themselves.[7]
The intruders have been divided into two categories according to this paper.[24] (External intruders, Internal Intruders). In this paper, the author uses an ANN as disconnected Ids to collect and analyze information from various elements of the Internet of Things, as well as to organize and recognize a DOS. According to a research paper, the Internet of Things is still in its infancy. However, it has attracted the attention of major corporations. That is why it is necessary to identify any illegal individuals or systems. IDS monitors the entire network, as well as all incoming and outgoing packets. It is being evaluated on its own capacity to prevent divided Denial of Service by the researchers, who have performed intrusion detection using Artificial Neural Networks that have been trained using web packet trace data.[5]
In accordance with this, the authors of this research have developed a new study that incorporates the multi-Layer design of the Internet of Things system. In-depth learning algorithms were then applied to the IoT organization to filter network information to group actions on the point of "normality" or, alternatively, "attacks" throughout the layers of the structure. KDD 99'Cup [8] has been used by experts.
Furthermore, in 2019, Li et al. proposed a method based on the fact that IoT information contains eradication in addition to other IDS, whereas urban areas are reliant upon deep learning. The testing demonstrated that the proposed framework exhibits a superior exhibition to conventional procedures. It reduces the bunching time by an appropriate amount; nonetheless, the grouping precision decreases when the framework is compacted.
Moreover, in 2019, Le et al. developed a new deep learning IDS. Brilliantly, a system was developed for the design of an element's interaction. During this phase, a drop in measurements resulted in the significant highlights for interruption detection, a subset of the first list of capabilities. The next step in the process was to create a large number of IDSs and then prepare them using the selected highlights.
It is not only IoT security that's at risk from infected Internet-connected devices but the entire Internet ecosystem as a whole. Since recently, security attack vectors have become more complex. As a result, it is imperative to upgrade security methods by implementing newer technologies. At the price of proper embedded security and verification, this paradigm helps expedite time-to-market. The Machine to Machine specifications recommends various security measures for IoT applications. Because of the fascinating discoveries that have already been demonstrated in the security

business, they have chosen to integrate machine learning techniques into our IDPS strategy.
Finally, in recent years, the Internet of Things (IoT) has emerged as the most rapidly emerging progressive invention, offering the ability to digitize and coordinate diverse businesses, resulting in enormous revenue opportunities and total GDP growth. The Internet of Things improves the coordination of all organizations, resulting in more significant data and information about tasks being collected. IoT is being used in a variety of projects to connect data, services, and individuals for creative activities in a variety of broad areas, including savvy power, smart cities, medical care, robotization, agribusiness, and transportation. IIoT is being used in a variety of projects to connect data, administrations, and individuals for creative activities in an assortment of board areas. The interruption recognition framework, which examines network data and analyzes network behavior, is perhaps the most important security solution for protecting IIoT applications from threats. According to what we learned in the last section, there are a variety of approaches to detect intrusions. Unlike deep learning, which involves continuous human observation, deep learning is the most efficient method. Even though Deep Learning Neural Networks have not been in the technological era for a very long time, they are becoming increasingly popular. It has advantages as well as disadvantages. When using a deep learning approach, it is required to train the Neural Network using a data collection of examples. It can be really challenging.
If a new threat emerges for which the network has not been taught, this can cause problems for the network. The Deep Learning technique is a highly smart approach in the subject of IDS, despite the fact that it has both positive and negative aspects. It is past time to conduct additional deep learning and machine learning research. As a result, it can detect intrusions more precisely and make the system more intelligent.

2.2 IoT Architecture

IoT architecture is built with a four-stage process where the data flows from the sensors, which are associated with the 'things' with the help of a network. These stages are Sensors/Actuators, Data acquisition system, Edge analytics, Cloud analytics [17]

- Sensors/ Actuators: "Internet of things" a thing can be embedded on actuators so that it can accept, emit and process a signal.
- Data Acquisition: In this part, the data acquisition system actually collects the data from sensors and starts to convert it from analog to digital. Also, it aggregates the data before sending it for the processing step.
- Boundary Analytics: The IoT info is aggregated and loaded. For processing the data, it will need to reduce the volume of it previously it goes to the info inside. This is the point when the boundary analytic works.
- Cloud logic: This primary and last process happens in the data center. Data uses higher in extent refinement to get expressed to any cloud-based system.

2.3 Important features for IOT solutions:

An Internet of Things system deals with large amounts of data. Therefore, a system that can be used on any sort of device and that is easy to use is essential. We came up with a few aspects that are essential for a simple yet effective Internet of Things system.

2.3.1 Light-Weight:

Many IoT devices are not suitable for preparing against malware programming because of their low-borderline operating scheme that is recycled like that of PCs and cell phones. There are no facilities that use advanced tactics to protect against the finer hazards of molecular software, and adequate mental space is required to hold the steadily rising amount of molecular software knowledge. Any bond revives, or gadget introductions can be facilitated by falling security measures, making it easier for security experts to do so and keep track of new organizations or other items that are expected to deal with the display. As a result, falling game plans produce a mechanism that is efficient, flexible, and capable of being executed in various ways.

2.3.2 IDS:

An Intrusion Detection System (IDS) is a security innovation that was developed primarily to identify weaknesses that were being exploited against a specific application or computer system. Interruption Prevention Systems (IPS) have expanded the capabilities of IDS by adding the ability to impede threats and distinguish them. As a result, IPS have become the predominant sending choice for IDS/IPS innovations and have become the predominant sending choice for IDS/IPS innovations. More specific information on the organizational structure and capabilities that characterize the IDS organization will be provided in this article. An IDS must be placed outside of the company's network in order to identify potential risks, which means that accurate data flow between the sender and recipient will be disrupted. Most IDS solutions query an inline traffic stream using a TAP or SPAN port to avoid interfering with normal network operations. In the beginning, IDS was developed in this manner because the depth of inquiry required for interruption identification could not be completed at a rate that could keep up with the speed at which elements of the organization's system exchanged information on a real-time basis.
Because of this, the IDS can also be used as a listening device. Monitoring of traffic is done by the IDS. It reports its discoveries to a system administrator, but it is powerless to prevent a recognized hacker from taking control of the system. After entering the organization, assailants are well-prepared to exploit holes as soon as they can, leaving the IDS with an inadequate arrangement for a countermeasure device. [33]

2.3.3 Multi-Layered:

The multi-sectoral system manages variance and, additionally, their information at many layers, resulting in an astonishing structure. In an IoT network, data is transmitted by a range of equipment, standardized, and stored in a number of ways, as well as moved to various locations. Solitary-overlay models may not result in enhanced performance in IoT architectures due to their domain or component level restrictions. However, a multi-layer system is communicated throughout the system, accepting cycles at various levels, from sophisticated to simple, depending on the scenario.

2.4 DEEP LEARNING

AI innovation is utilized in a variety of areas of contemporary life, ranging from web searches to content separation on social networks to suggestions on business websites. It is becoming increasingly prevalent in consumer products such as cameras and cell phones. Artificial intelligence frameworks are used to recognize objects in photographs, decode spoken language into text, match news items, posts, or commodities with customers' preferences, and select the most relevant results from searches. Progressively, these applications make use of a class of methods known as profound learning to accomplish their goals. In its primitive structure, regular artificial intelligence procedures were limited in their ability to deal with routine information. In the past, developing an example recognition or artificial intelligence framework necessitated the cautious design and significant space expertise to plan a feature extractor that transformed basic information (for example, the pixel values of a picture) into an appropriate inner portrayal or element vector from which the learning subsystem, frequently a classifier, could distinguish or order designs in the information.[34] Portrayal learning is a collection of approaches that enable a machine to be dealt with rudimentary knowledge and, as a result, identify the portrayals required for location or characterization.[30] Profound learning techniques are portrayal learning techniques that have varying degrees of representation. They are acquired through the creation of straightforward yet non-direct modules that each change the portrayal at one level (beginning with the crude information) into a portrayal at a higher, somewhat more conceptual level (beginning with the crude information). Extremely sophisticated abilities can be taught if the structure of a sufficient number of such changes is followed closely enough. Higher levels of depiction boost parts of the information that are important for separation and suffocate immaterial variants when it comes to ordering assignments. For example, an image can be represented by a range of pixel values. The learned elements in the first layer of depiction are often concerned with the presence or absence of edges in specified directions and sections of the picture. It is customary for the second layer to separate themes by identifying unique edge plans and paying little attention to minor variations in edge placements. It is possible that the third layer may group themes into larger mixes that are related to different elements of natural articles and that subsequent layers will distinguish things as blends of these parts. This is an important aspect of profound learning since it means that these layers of elements are not intended by human specialists but rather are obtained from knowledge by employing a generally beneficial learning approach. Applied profound learning is making substantial strides in the treatment of difficulties that have thwarted the best efforts of the man-made reasoning community for a very long time. Eventually, it will be capable of finding many-sided structures in densely layered information, and it will be beneficial in a wide range of sectors in science, commerce, and government, as well as in other fields. Not only has it broken world records in picture recognition 1–4 and discourse recognition5–7, but it has also outperformed other artificial intelligence methods in predicting the activity of potential medication molecules8, dissecting atom smasher data9, 10, reproducing mind circuits11, and foreseeing the effects of mutations in non-coding DNA on quality articulation and disease. 12,13. Perhaps even more unexpected, profound learning has produced astonishingly optimistic outcomes for a variety of tasks in normal language understanding14, including topic organization, opinion exploration, question answering15, and language translation16. We anticipate that profound learning will achieve significantly more significant accomplishments in the not-too-distant future because it requires little to no manual design and can thus take advantage of increases in the amount of available computing and information in the near future. The development of new learning algorithms and models for profound neuronal organizations, which are now being developed, will only serve to speed this progress. [16]

2.5 Algorithm

In our thesis work, we developed three methods to test our theory and provide evidence to back it up. The results were also calculated individually for each algorithm, which significantly improved. The most commonly used algorithms are RNN, LSTM, and GRU. It is possible to think of algorithms as a collection of efficient stages that are used to solve problems such as processing, data preparation, and default advising. The algorithm is also a powerful technique that can be shown in a short period of time, which is particularly advantageous. Nevertheless, another significant point is that such algorithms are the most effective means of discovering the optimal solution to a particular problem most straightforwardly and practically. We will get results for the learning rate, hidden layers, and time-steps if we run numerous test-train iterations. Then, with the use of an evaluation matrix, we will gather information regarding perfection, precision, recall, and false positives.

2.5.1 Recurrent Neural Networks (RNN)

Recurrent neural network (RNN) is a magnificent and powerful sort of neural structure, with an environment rich in encouraging data, which is used to justify the notion that they possess internal memory. RNNs have risen to the top of the priority list as a result of the development of LSTM, as well as the increase in computer power and the massive amount of information technology we must now work on.[13] An example of a typical RNN structure is shown in the following example:

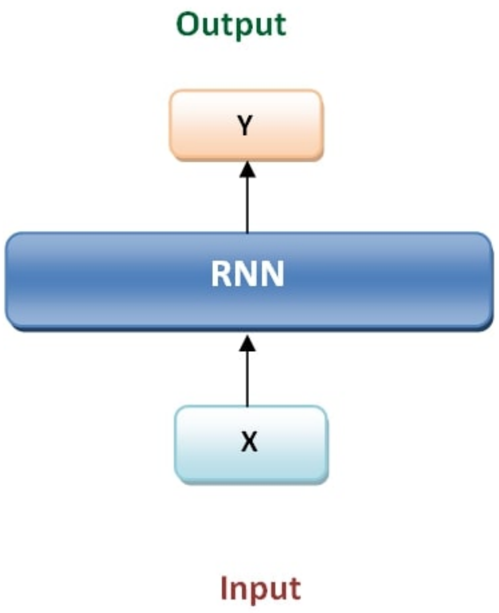


Figure 2.1: Rnn Diagram

The RNN contains two types of information sources: current and historical. This is vital since the data collection contains critical information about what is likely to occur in the near future, which is why the RNN is capable of performing jobs that other statistical methods are unable to complete successfully. Finally, the feed-forward neural structure selects a weight grid in its retrospective components, just as it does with other in-depth research statistics, and produces the desired outcomes. Furthermore, the recurrent neural organization will alter your loads of both angle reduction and backward distribution over time as a result of your training. [18]

2.5.2 Long Short-Term Memory (LSTM)

LSTM is a neural network that is similar to RNN. The LSTM consists of four neural networks that work together. It is made up of different types of memory blocks, which are referred to as cells. The fact that the LSTM is familiar with the work of the news channel team, which includes the murder case, demonstrates the utility of the tool. [3]

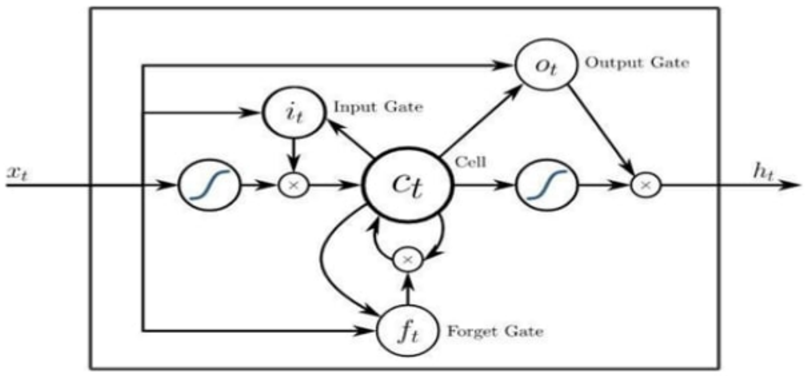


Figure 2.2: LSTM Diagram

Data is stored in cells and memory management is done through 3 gates. The three gates are –

1. Forget gate:
The first gate is referred to as the Gate of Forgetting. This gateway determines what information will be discarded when we remove the cell from its current state. [23] This is determined by the first layer of sigmoid, which considers the previous effect as well as the current input, which is:

$$ft = \sum(Wf * [ht - 1, xt] + bf)$$

2. Input gate: Another gate is used to insert another sigmoid layer, which draws the numbers between 0 and 1 and determines its result. In addition, those values will be re-established. The election values that will be used to restore cell status are calculated based on the value of the tanh layer, and these two values are combined to create a review for the entire province. [23]

$$it = \sum(Wi * [ht - 1, xt] + bi)$$
$$C't = tanh(Wc * [ht - 1, xt] + bc)$$

In the meantime, the old cell's status must be changed to that of a new cell. As a result, even though the previous steps have already determined what we will do, we still need to carry out the plan. [15]

$$Ct = ft * Ct - 1 + it, xt * C't$$

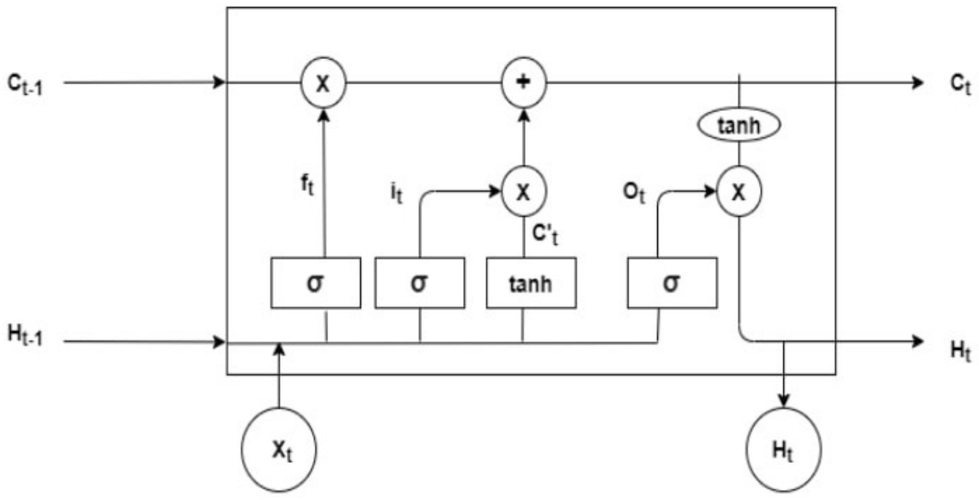


Figure 2.3: Forget Gate

3. Output Gate:
We must make a decision on what to make public. Despite the fact that this output will be based on the current status of our cell, it will be of the filtered type. As an initial step, we employ the Sigmoid Layer algorithm to determine which portions of the cell need to be released. As a result of this, we place the cells in the desired positions using a tan layer with values ranging between -1 and 1 and multiply the result by the number of segments we have decided to make by releasing the sigmoid gate. According to statistics, the situation appears to be as follows:

$$ot = \sum(Wo * [ht - 1, xt] + bo)$$
$$ht = ot * tanh(Ct)$$

Moreover, LSTMs look very scary when we look at stats or cells alone. So, walking through them step by step has made it easier for them to understand. [23]

2.5.3 Gated Recurrent Units (GRU)

Over the past few years, repetitive neural networks with repetitive hidden units have shown promising results in a variety of programs and applications. Gated Recurrent Units (GRUs) are a type of unit that is frequently repeated and is one of the most closely related varieties among the most frequently repeated units.[22]
GRUs can be thought of as a more straightforward variant of LSTMs, which is also true. The GRU unit, which was introduced in 2014 and is said to have been inspired by the Long Short-Term Memory unit, was introduced in 2014. In any case, the former is significantly easier to conceptualize and implement in models. [25]
The GRU's general organizational diagram is as follows:

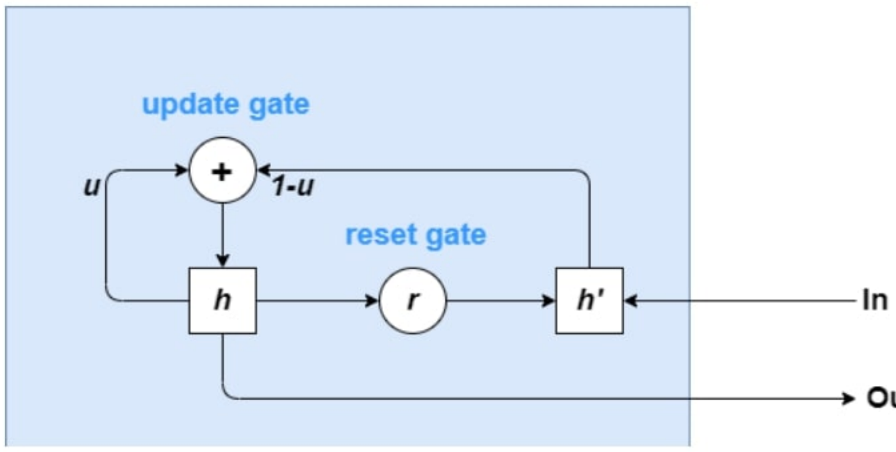


Figure 2.4: Gated Recurrent Unit

A variation on the intermittent neural organization, gated repetitive unit networks can deal with recollections of consecutive information by storing past inputs in the interior condition of organizations and planning from the historical backdrop of past contributions to target vectors on a fundamental level.
Furthermore, two entryways are provided in GRU, including a reset door that changes the fuse of new contributions with the past memory and an updated door that controls the protection of the valuable memory. Each secret unit's ability to recall or neglect information while perusing or producing an arrangement is adaptively controlled by the reset entryway and the updated door. As a side note, the GRU's operations continue to such an extent that when the reset entryway is close to zero, the secret state is compelled to disregard the previous secret state and is reset with the most recent information available. [27] The amount of information from the previous secret state that is transferred to the current secret state is controlled by the updated door. This cycle functions similarly to the memory cell in the Long Short-Term Memory organization and aids RNN in retrieving long-term data from memory.[26]
Finally, GRU can store and filter information through the use of their own gateways for refresh and reset, which they have developed. The problem of the perishable gradient is eliminated as a result of the fact that the model does not always wash out new inputs but rather stores the correct information and transfers it to the next network step instead. In addition, it can be used to improve the memory capacity

of a recurrent neural network and provide simple model training to the network. If correctly trained, it has the potential to be incredibly beneficial in even the most difficult of circumstances.[10]

2.5.4 Fold Cross Validation:

With cross-validation, you can evaluate machine learning models on a tiny sample of data, similar to how you would with resampling. The process includes only one parameter, k, which specifies the number of groups into which a given data sample should be divided. In order to distinguish it from other cross-validation techniques, the method is commonly referred to as k-fold cross-validation. When a precise value for k is specified, it can be substituted for k in the model's reference, for example, k=10 for 10-fold cross-validation.
When it comes to machine learning, cross-validation can be used to determine how well a model can adapt to new data. In other words, a tiny sample will be used to see how well the model performs when predictions are made using data that was not included during the model's training. It is a popular strategy since it is straightforward to grasp and produces a less biased or optimistic estimate of model competence than other approaches, such as a simple train/test split.
The procedure can be summed up as follows:

- Using a random number generator, shuffle the dataset.
- Sort the data into k groups and then sort the groups again.
- Write the following for each distinct group.
- The group can be used as a holdout or test data set.
- Use the remaining categories as a training data set for your students.
- Testing a model against the test set is distinct from fitting a model to the training set.
- Keep the evaluation score, but throw out the model entirely.
- Create a summary of the model's ability based on the sample of model evaluation scores.

It is critical to note that each observation in the data sample is assigned to a specific group and remains in that group throughout the technique. This means that each sample has a chance to be used in the holdout set once and to be used to train the model a total of k times. [9]

Chapter 3

Comparative Study of Deep Learning:

3.1 Methodology

Firstly, we have imported the KDD-99 cup and NSL-KDD datasets and converted the string data types to integer data types. Following that, we standardized the values of the datasets. Then we divided the dataset into two parts: training and testing (in 70:30 ratio). After that, we trained the dataset using LSTM, Vanilla LSTM, Stacked LSTM, Bidirectional LSTM GRU, and RNN algorithms. We incorporated the dataset into each algorithm in three different ways. First, we have just divided the dataset into training and testing subsets, which will be used in the algorithms. Then, we utilized repeated k-folds, in which we divided the dataset into 5 folds and repeated the process for 10 times. After that, we used feature selection to identify the top 10 features, which we then put into algorithms. We have used this alternative processing of the datasets to determine how the algorithm works for each of these different representations.

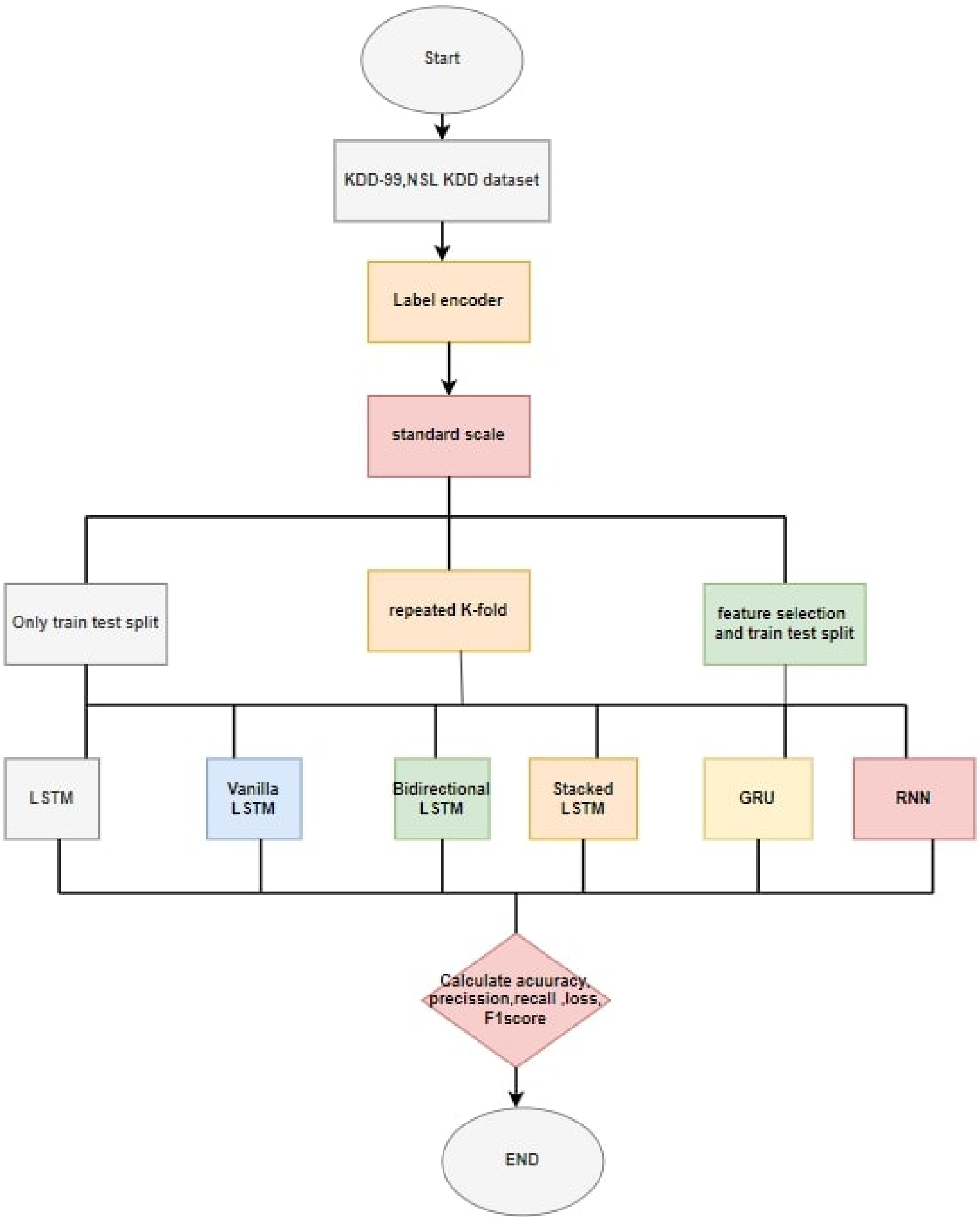


Figure 3.1: Flowchart of Work Plan

3.2 Dataset description:

DARPA is a harsh dataset that serves as a starting point. The DARPA dataset KDD99 is a part-disengaged variant of the DARPA dataset. The NSL-KDD is the variation of the KDD99 that corresponds to the NSL.

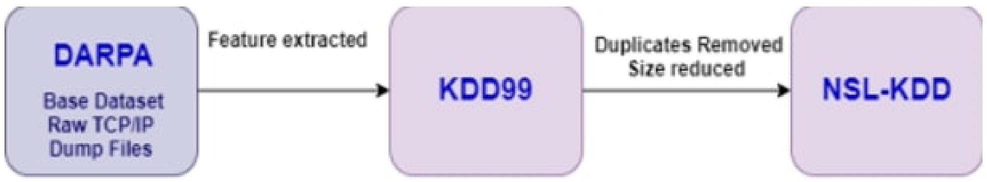


Figure 3.2: The Relation Between Dataset

Name	Training Size	Testing Size	Note
DARPA 99	6591458 kb (6.2 gb)	3853522 kb (3.67 gb)	Base Dataset, RAW TCP/IP Dump Files
KDD99	4898431	311029	Features extracted and preprocessed for machine learning
NSL-KDD	125973	22544	Duplicates removed, size reduced

Table 3.1: Dataset information

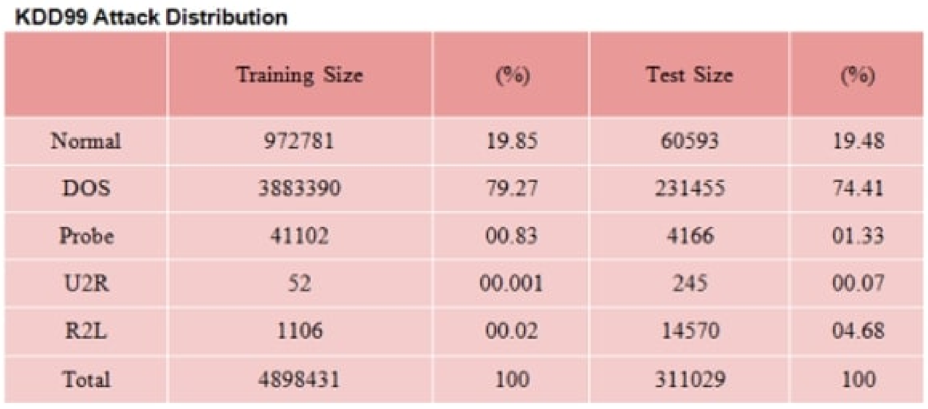
3.2.1 KDD99:

The dataset KDD99, which was created in the year 1999 and is the most widely used in the field of IDS, is the most popular. [6] The interruption identification datasets from KDD ninety-nine, which are based on the office ninety-eight dataset, provide marked data to scientists working in the field of interruption identification, and because it is marked data, it is freely available. Diverse specialists used the datasets in the KDD ninety-nine interruption identification competition to target the employment of artificial intelligence for interruption identification and proclaimed location rates up to ninety-one with phony positive rates under the I Chronicles competition rules. [2]

KDD99 has following qualities:
1. It contains twenty-four assault types in preparation and fourteen plus assault types in testing, for a total of thirty-eight attacks on the data set. These fourteen new assaults are hypothetical tests of an IDS's ability to conceal attacks and, as a result, to obscure attacks. At the same time, it is difficult for the Artificial Intelligence-based Detection Unit to distinguish between these fourteen new attacks on the network.
2.Kdd99 is a dataset that is energetically imbalanced and can be used to attack events. Attack traffic accounts for nearly eighty percent of the stream's total volume (3925650 attack models through and through 4898430 cases). A typical association consistently contains approximately 99.99 percent of conventional events per penny of its total number of events. This rule is mishandled by KDD99.

3. It is a huge dataset for most AI calculations:

The primary collection (fundamental) contains nine highlights that include fundamental information, such as the convention, administration, and term, among other things. The second group (content) comprises thirteen elements that contain information about the substance, such as the login exercises, for example. For example, the number of associations that are identified with a similar host within a two-


KDD99 Attack Distribution

	Training Size	(%)	Test Size	(%)
Normal	972781	19.85	60593	19.48
DOS	3883390	79.27	231455	74.41
Probe	41102	00.83	4166	01.33
U2R	52	00.001	245	00.07
R2L	1106	00.02	14570	04.68
Total	4898431	100	311029	100

Figure 3.3: KDD99 attack distribution

Group	Gathering/Features	Count
Basic	f1-1,f1-2,f1-3,f1-4,f1-5,f1-6,f1-7,f1-8,f1-9	9
Content	f1-10,f1-11,f1-12,f1-13,f1-14,f1-15,f1-16,f1-17,f1-18,f1-19,f1-20,f1-21	13
Tune	f1-23,f1-24,f1-25,f1-26,f1-27,f1-28,f1-29,f1-30,f1-31	9
Host	f1-32,f1-33,f1-34,f1-35,f1-36,f1-37,f1-38,f1-39,f1-40,f1-41	10

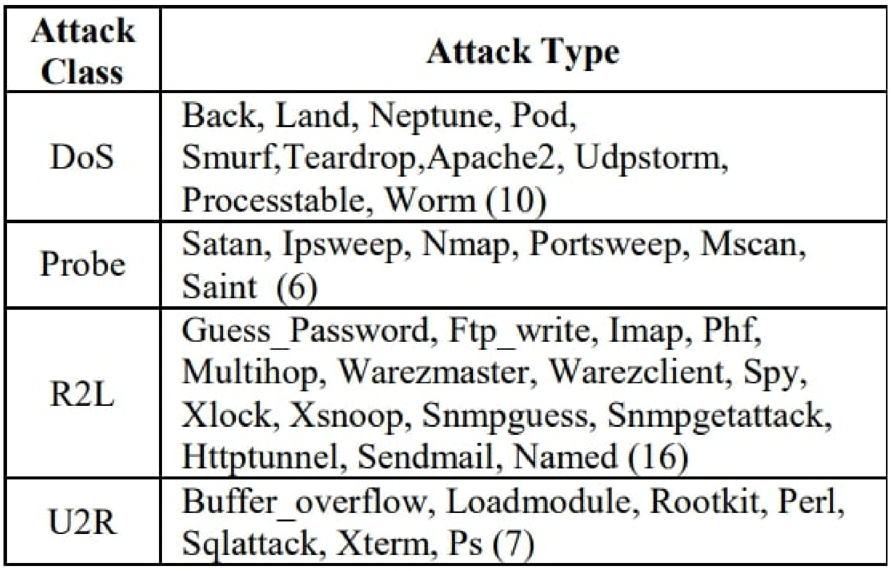
Table 3.2: Four gatherings of provisions in KDD99

second time frame is provided by the third gathering (time), which contains nine time-sensitive elements. The fourth (have) section contains ten have-based provisions that provide information about the association with the host, for example, the rate at which associations with a similar objective port number are attempting to be accessed by various hosts are attempting to get to the host. [14]
The KDD dataset contains twenty-four different assault types. There are forty-one elements that are addressed in the process of preparing for and testing exams. Their names are given to all things that are considered to be "ordinary" or "assault-type." Parts can be divided into three categories in this section. Fundamental social affair depicts the components that are used for providing information on the request that is utilized as affiliations, second assembling of components depicts judgment orders, and third assembling depicts appearances that pass on information around affiliations that have a similar goal among the same organization, and the third assembling depicts appearances that pass on information around affiliations that have a similar goal among the same organization.

3.2.2 NSL-KDD:

NSL-KDD includes similar components as KDD99, where it includes forty-one elements and one category of quality. The evaluation appraisal for the peculiarity area totally relies upon a few AI techniques upon various planning and testing datasets. The genuinely monstrous deficiency in the KDD instructive archive is the colossal number of disheartening documents for almost 78%, along with 75% replicated in train and test set, autonomously.
In the NSL-KDD data set, the attack classes that have been identified are divided into four categories. DOS: When a victim's resources are depleted, he or she is

unable to respond to valid requests. For example, flooding the victim’s system with requests can result in a denial of service assault on the victim. Therefore, the following are relevant characteristics: ”source bytes” and ”percentage of packets with errors.”
Probing: The goal of surveillance and other probing attacks, such as port scanning, is to gather information about the remote victim. ”Duration of connection” and ”source bytes” are two important characteristics to consider.
U2R: Unauthorized access to local super user (root) privileges (U2R) is a type of attack in which an attacker logs into a victim system using a normal account and attempts to gain root/administrator privileges by exploiting a vulnerability in the victim, such as a buffer overflow attack.” number of file creations” and ”number of shell prompts invoked” are two important metrics to track.
R2L: Remote-to-local (R2L) access is when an attacker intrudes into a victim’s computer from a remote location. For example, password guessing. The following are important characteristics: ”Duration of connection” and ”service requested” are network-level characteristics, while ”number of failed login attempts” is a host-level characteristic.

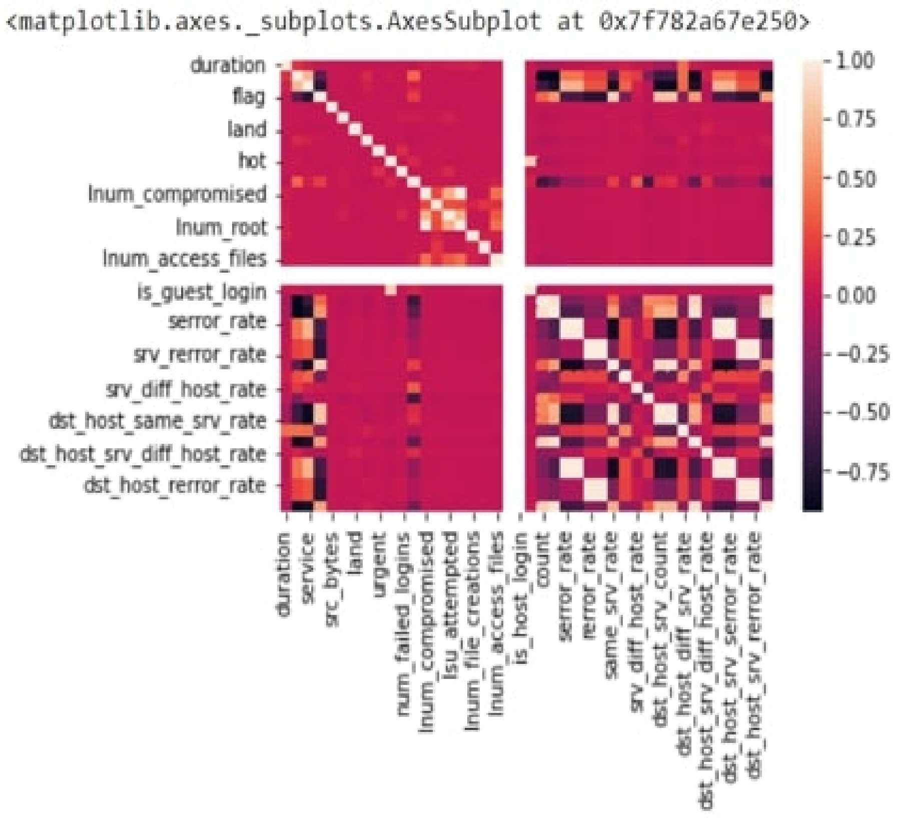


Attack Class	Attack Type
DoS	Back, Land, Neptune, Pod, Smurf,Teardrop,Apache2, Udpstorm, Processtable, Worm (10)
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint (6)
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named (16)
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps (7)

Figure 3.4: Broken down by attack and normal data distribution in NSL-KDD

3.3 Preprocessing of KDD-99 and NSL-KDD dataset:

We have taken KDD-99 cup and NSL-KDD dataset and processed the datasets for further use. Then we checked and removed if there is any null value. Then we checked and removed if there is any duplicate value. After that, there were features like “protocol type”, “service”, “flag” and “label” had data type object and other features had data type int. We changed the object data type to int. Furthermore, there were 42 columns. We divided the first forty-one columns as features and rest one as label. Then, we have used StandardScaler for scaling the dataset transformed the data such that its distribution will have a mean value 0 and standard deviation of 1. Then the datasets will be separated into training and testing for training and testing purpose and we set the training set 70% and testing set 30%.We have used K-foldcross validation for using in algorithms. Moreover, we have also used features selection for selecting best 10 features. We have reshaped the dataset 2d to 3d for using them in LSTM, GRU and RNN. After that, we calculated the data correlation and plotted a graph for KDD-99cup dataset and NSL KDD dataset in figure 8 and figure 9:



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f782a67e250>
```

Figure 3.5: Data co-relation of data in KDD-99

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2e9ffc6a90>
```

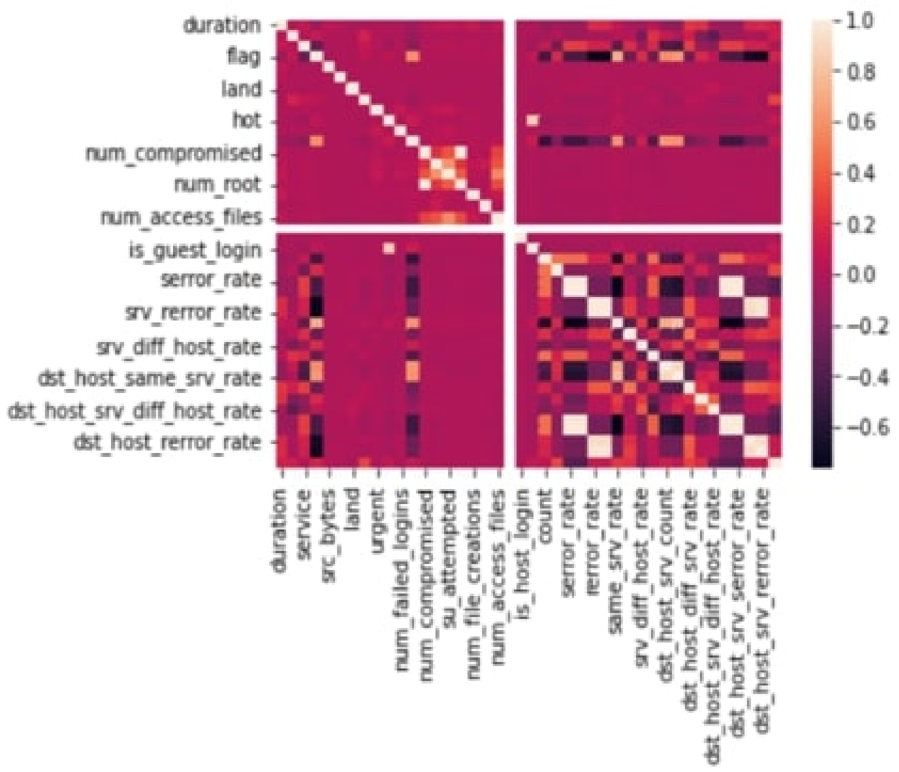


Figure 3.6: Data co-relation of data in NSL-KDD

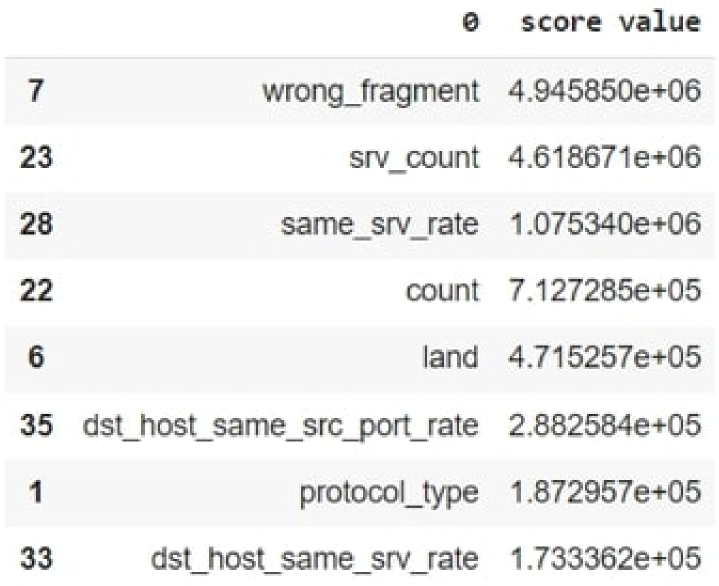
We also processed the KDD-99 and NSL KDD datasets for three different tasks, including using the dataset divided into a 70:30 ratio, using fold cross validation, and applying in feature selection (see Additional Resources).

Train test split: We obtained accuracy, precision, recall, and f1-score of 0.566 by dividing the KDD-99 dataset into a 70:30 ratio (70 percent data for training and 30 percent data for testing). As an added bonus, we didn't have to divide the NSL KDD dataset because we had two separate files for training and testing. We obtained accuracy, precision, recall, and a f1-score of 0.002 as well as precision and recall.[21]

Fold cross validation: For the fold cross validation, we utilized Repeated KFold and made 5 folds, which we then repeated for a total of 10 times. After doing this fold cross, we obtained accuracy, precision, recall, and a f1-score of 0.95 for the KDD-99. The accuracy, precision, recall, and f1-score for NSL KDD were all excellent, as was the recall.

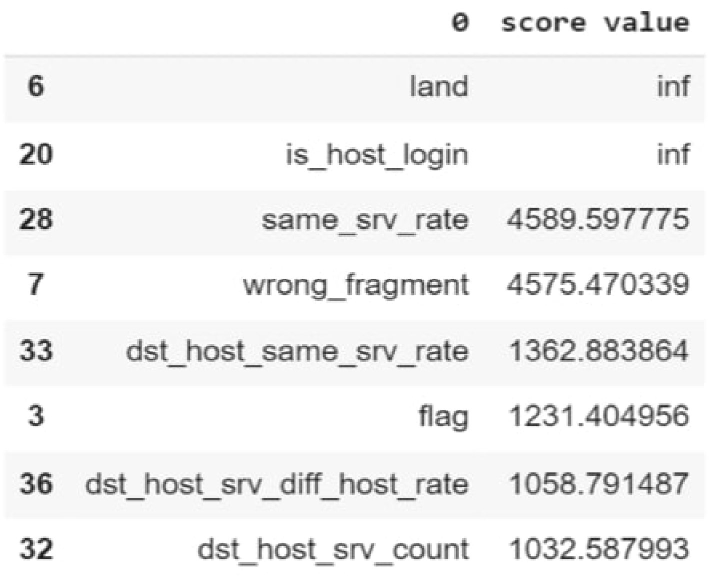
Feature selection: By applying the feature selection procedure to the dataset, we were able to identify the best 8 features from among 41 possible features, which are depicted in the figure along with their co-relation.

Finally, we discovered that KDD 99 had accuracy, precision, recall, and a f1-score of 0.460, while NSL KDD had accuracy, precision, recall, and a f1-score of 0.010, respectively. The finest eight aspects of KDD99 and NSL KDD, as well as their scores, are depicted in the figure. It takes significantly less time to use feature selection than it does to use other ways.



	0	score value
7	wrong_fragment	4.945850e+06
23	srv_count	4.618671e+06
28	same_srv_rate	1.075340e+06
22	count	7.127285e+05
6	land	4.715257e+05
35	dst_host_same_src_port_rate	2.882584e+05
1	protocol_type	1.872957e+05
33	dst_host_same_srv_rate	1.733362e+05

Figure 3.7: Best features of KDD99



	0	score value
6	land	inf
20	is_host_login	inf
28	same_srv_rate	4589.597775
7	wrong_fragment	4575.470339
33	dst_host_same_srv_rate	1362.883864
3	flag	1231.404956
36	dst_host_srv_diff_host_rate	1058.791487
32	dst_host_srv_count	1032.587993

Figure 3.8: Best Features of NSL KDD

Chapter 4

Experimentation

As part of this experiment, we have implemented the LSTM algorithm on the KDD-99 cup dataset as well as the NSL-KDD dataset. We used feature selection, fold cross validation, and train test split to determine which approach produces the best results on each dataset. In each algorithm, we employed a total of 10 epochs and a batch size of 32.
Our hidden layers have sizes of 268, 128 and 64 pixels, respectively, and our activation function is relu. We have employed three hidden layers in MLP, with each layer having a size of 268, 128 and 64 pixels, respectively. In addition, our random state is one.
LSTM is composed of two input LSTM network layers, each of which has a size of 128. The second LSTM network layer has a size that is the same as the first. Following that, we have one hidden layer with a dense size of 30 and finally one output layer with a size of 30.
One input GRU network layer with a size of 128 and another GRU network layer with a size of the same as the previous one make up the GRU network layer hierarchy. Following that, we have one hidden layer with a dense size of 30 and finally one output layer with a size of 30.
In RNN, we have one input layer with a size of 41 pixels, two simple RNN layers with a size of 512 pixels, and finally a dense layer with a size of 30 pixels.

4.1 Performance results of applying LSTM on KDD 99:

By relying simply on train test split, we were able to achieve an accuracy of 82% with a loss of 0.48 percent. Following that, we obtained 96% accuracy with a loss of 0.27% by utilizing only k-fold cross validation. We then achieved 77 percent training accuracy with a loss of 3.20% by relying solely on feature selection.

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.82	0.99	0.82	0.89	0.48	45 min
20	0.82	0.99	0.82	0.89	0.49	94 min
50	0.81	0.99	0.81	0.88	0.30	263 min

Table 4.1: Performance result of LSTM on KDD99

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.77	0.99	0.77	0.86	3.20	20 min
20	0.81	0.99	0.81	0.89	4.38	41 min
50	0.80	0.99	0.80	0.88	6.77	86 min

Table 4.2: Feature Selection Method

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.96	0.99	0.96	0.97	0.27	45 min
20	0.99	1.00	0.99	0.99	0.10	94 min
50	0.99	1.00	0.99	0.99	0.11	263 min

Table 4.3: Fold Cross Method

4.2 Using Difference Types of LSTM to compare the result

Algorithm	Epoch	Loss	Accuracy	Precision	Recall	F1 Score
Vanilla LSTM	10	2.70	0.61	0.95	0.61	0.71
	20	0.92	0.89	0.75	0.79	0.79
Stacked LSTM	10	0.44	0.79	0.96	0.79	0.85
	20	0.48	0.92	0.98	0.92	0.94
Bidirectional LSTM	10	3.30	0.75	0.99	0.75	0.83
	20	0.55	0.71	0.97	0.71	0.81

Table 4.4: Test-Train Split Method

Algorithm	Epoch	Loss	Accuracy	Precision	Recall	F1 Score
Vanilla LSTM	10	1.99	0.89	0.96	0.89	0.92
	20	1.23	0.83	0.96	0.86	0.96
Stacked LSTM	10	0.32	0.95	0.99	0.95	0.96
	20	0.32	0.91	0.97	0.91	0.94
Bidirectional LSTM	10	0.24	0.97	0.99	0.97	0.98
	20	0.13	0.98	0.99	0.98	0.99
	20	0.55	0.71	0.97	0.71	0.81

Table 4.5: Fold Cross Method

4.3 Performance results of applying GRU on KDD 99:

By adopting a simple train-test split, we were able to achieve training accuracy of 99.87% with a loss of 0.0054 and testing accuracy of 86.03% with a loss of 1.5273 while minimizing errors.
As a result of using only k-fold cross-validation, we were able to get a training accuracy of 99.87% with a loss of 0.0057 and a testing accuracy of 42.93% with a loss of 7.35. In our experiments, we were able to achieve training accuracy of 98.92% with a loss of 0.428% and testing accuracy of 47.06% with a loss of 4.87% by utilizing only feature selection techniques.

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.95	0.99	0.95	0.94	0.20	49 min
20	0.83	0.99	0.83	0.90	5.34	96 min
50	0.85	0.91	0.85	0.91	2.41	225 min

Table 4.6: Test-Train Split Method

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.46	0.54	0.46	0.49	8.22	19 min
20	0.47	0.54	0.47	0.49	8.87	39 min
50	0.77	0.90	0.77	0.80	3.25	153 min

Table 4.7: Feature Selection Method

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.98	0.98	0.98	0.98	0.90	61 min
20	0.98	0.97	0.98	0.97	0.47	102 min
50	0.98	0.98	0.98	0.98	0.35	192 min

Table 4.8: Fold Cross Method

4.4 Performance results of applying RNN on KDD 99:

Through the use of a simple train-test split procedure, we were able to obtain training accuracy of 99.93% with a loss of 0.0079 and testing accuracy of 85.49% with a loss of 12.193%. With only k-fold cross validation, we were able to achieve 99.93% training accuracy while incurring 0.0071% loss, and 99.795% testing accuracy while incurring 7.35% loss, which was sufficient for our purposes. Our training accuracy was 98.88% with a loss of 0.0427%, and our testing accuracy was 85.7% with a loss of 0.23% as a result of relying exclusively on feature selection.

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.80	0.99	0.80	0.88	8.86	65 min
20	0.80	0.99	0.80	0.87	6.48	140 min
50	0.86	1.00	0.86	0.91	4.78	370 min

Table 4.9: Test-Train Split Method

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.84	0.99	0.84	0.90	5.95	35 min
20	0.84	0.98	0.84	0.90	1.98	65 min
50	0.84	0.96	0.84	0.89	5.15	175 min

Table 4.10: Feature Selection Method

Epoch	Accuracy	Precision	Recall	F1 Score	Loss	Time
10	0.97	0.98	0.97	0.98	0.51	62 min
20	0.93	0.97	0.93	0.95	1.01	160 min
50	0.61	0.87	0.61	0.65	5.47	385 min

Table 4.11: Fold Cross Method

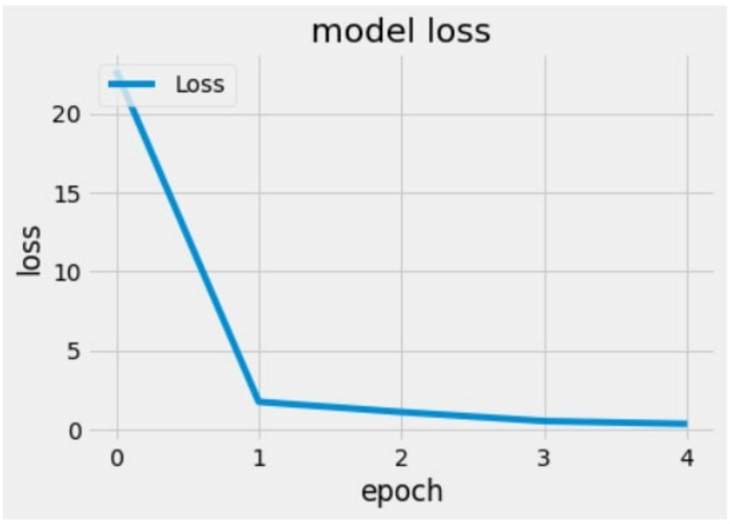


Figure 4.1: Model loss of LSTM using Test-train Split

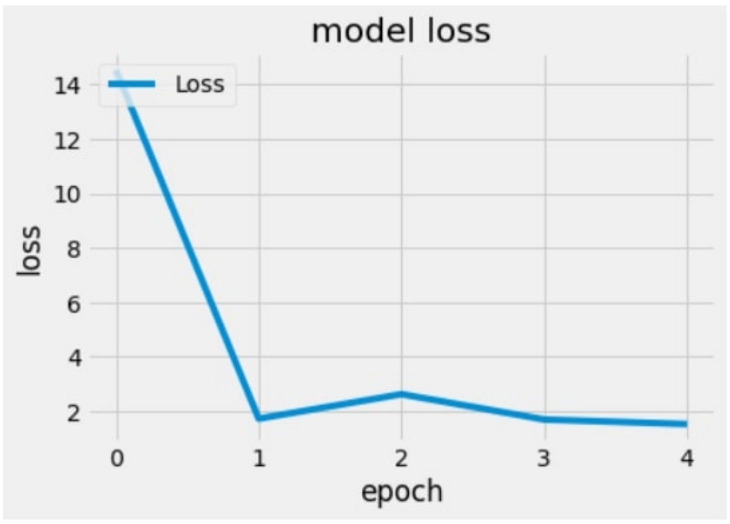


Figure 4.2: Model loss of LSTM using Feature Selection

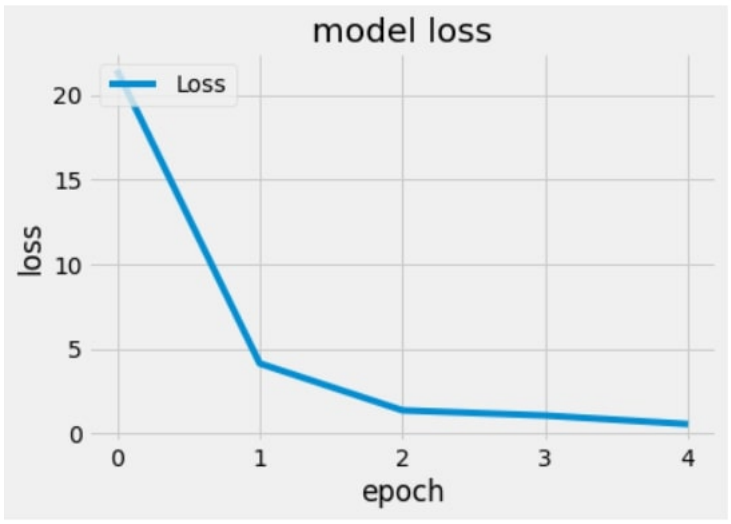


Figure 4.3: Model loss of LSTM using Fold Cross

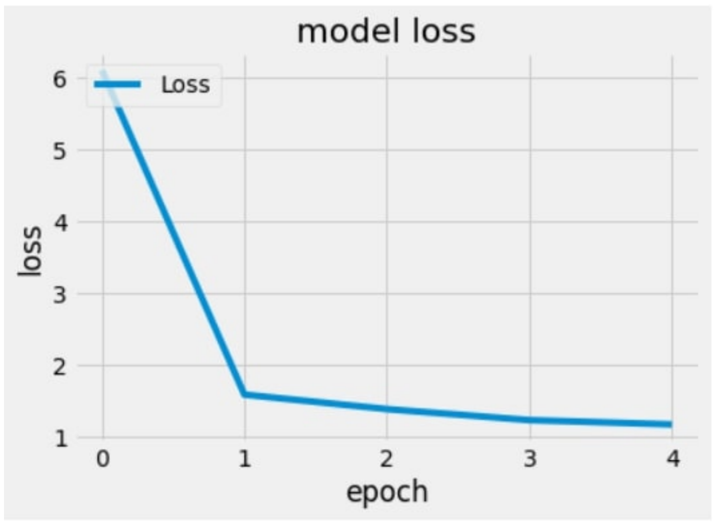


Figure 4.4: Model loss of GRU using Test Train Split

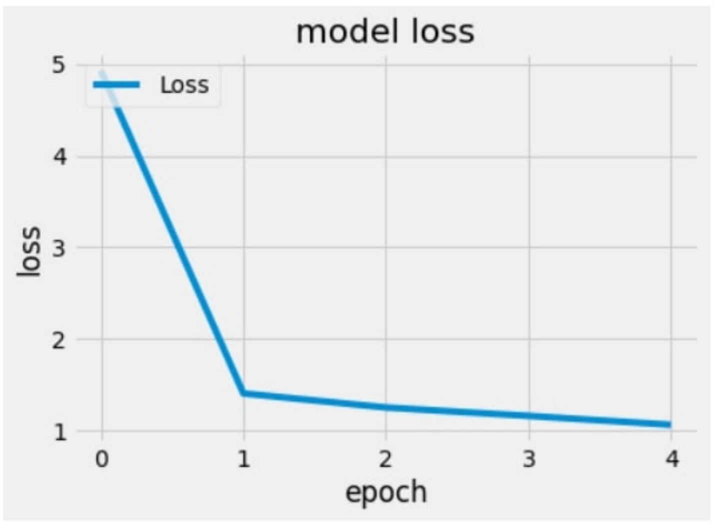


Figure 4.5: Model loss of GRU using Feature Selection

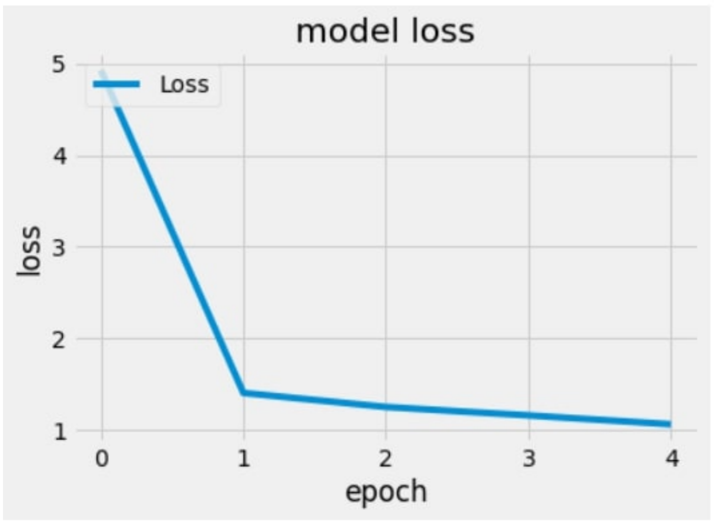


Figure 4.6: Model loss of GRU using Fold Cross

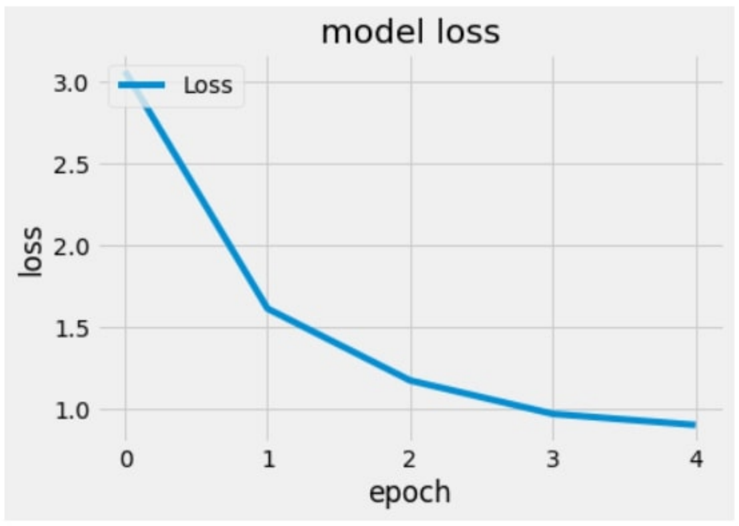


Figure 4.7: Model loss of RNN using Feature Selection

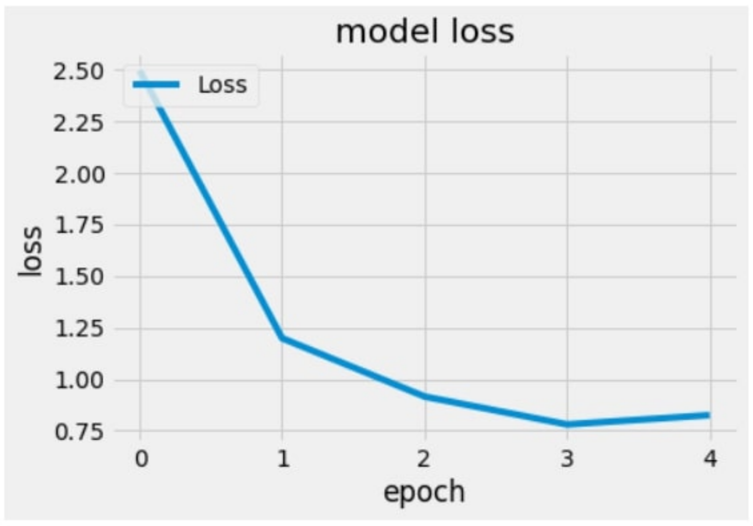


Figure 4.8: Model loss RNN using Fold Cross

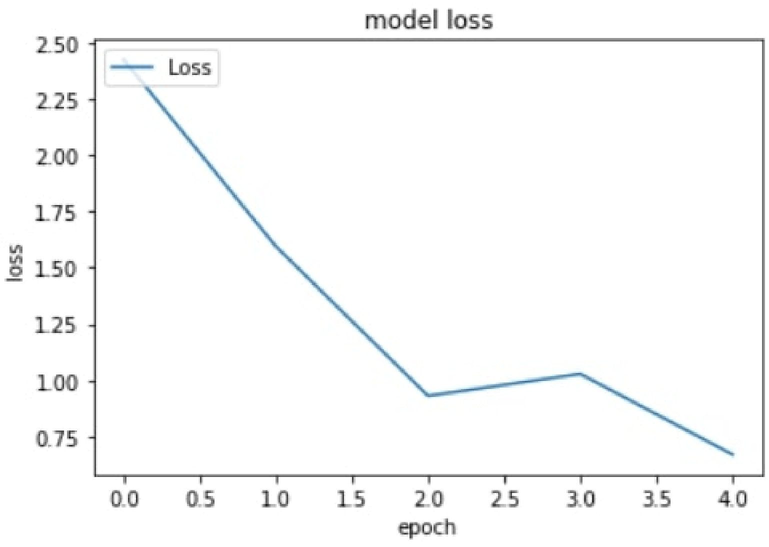


Figure 4.9: Model loss of RNN using Test-Train Split

4.5 Performance results of applying LSTM and GRU both on KDD:

We have used Bidirectional LSTM and GRU both. Test-train gives accuracy with 65%, where Fold cross give with 97% accuracy

Method	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Loss**	**Time (10 epochs)**
Test-train Split	0.65	0.79	065	0.68	1.29	212 min
Fold Cross	0.97	0.96	0.97	0.96	1.41	242 min

Table 4.12: Performance results of applying LSTM and GRU both on KDD

4.6 Performance results of applying LSTM, GRU and RNN on NSL-KDD:

Algorithm	**Method**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
LSTM	Test-Train Split	0.61	0.57	0.61	0.74
	Fold Cross	0.79	0.92	0.79	0.84
GRU	Test-Train Split	0.31	0.76	0.31	0.33
	Fold Cross	0.72	0.86	0.72	0.77
RNN	Test-Train Split	0.71	0.86	0.71	0.77
	Fold Cross	0.81	0.87	0.81	0.83

Table 4.13: Performance results of applying LSTM,GRU and RNN on NSL-KDD

Chapter 5

Result

5.1 Accuracy on KDD-99 Dataset:

If we look at the training accuracy of the algorithms in KDD-99 dataset we get

Method	LSTM	GRU	RNN
Train test split	0.81	0.95	0.80
Fold cross validation	0.98	0.92	0.85
Feature selection	0.96	0.48	0.84

Table 5.1: Accuracy on KDD-99 Dataset

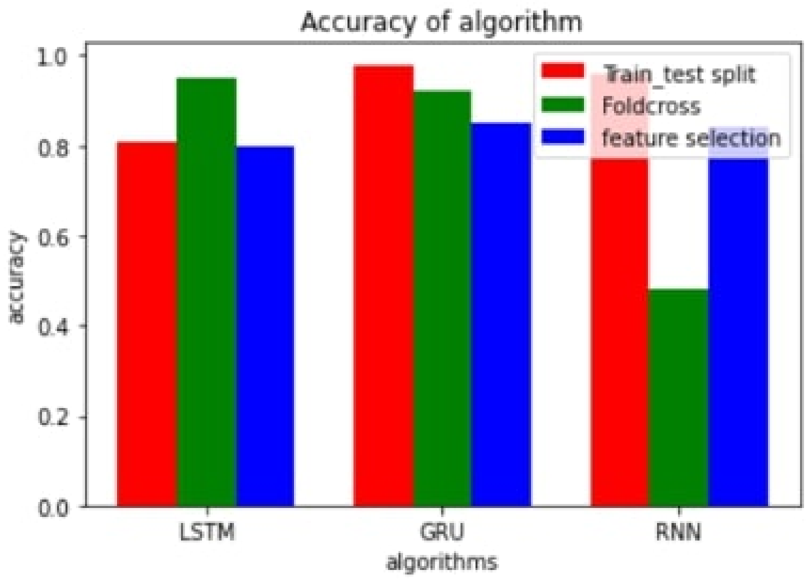


Figure 5.1: Bar diagram of training accuracy on KDD-99 dataset

Among algorithms like LSTM, GRU and RNN on an average LSTM gave best accuracy. It gave more that 95% accuracy in fold cross and feature selection and gave more than 80% in only train test split. By using Fold-cross method gave better result than other two methods and it performed best on LSTM.

Method	**Algorithm**	**Epoch**	**Accuracy**
Feature Selection	LSTM	10	0.77
”		50	0.80
”	GRU	10	0.46
”		50	0.77
”	RNN	10	0.84
”		50	0.84

Table 5.2: Accuracy Based on Feature Selection for 10 and 50 Epochs

We can observe from the Accuracy chart that the Feature Selection approach did not outperform the others.This is because, due to the intrinsic complexity of neural networks, the Feature Selection technique is inapplicable. On the other side, increased training time has the potential to improve results.
Furthermore, we obtained superior results by employing the Fold Cross approach. On the same data, we ran three different types of LSTM experiments.

Algorithm	**Method**	**Epoch**	**Accuracy**
Vanilla LSTM	Fold Cross	10	0.89
	”	20	0.83
Stacked LSTM	”	10	0.95
	”	20	0.91
Bidirectional LSTM	”	10	0.97
	”	20	0.98

Table 5.3: Accuracy Based on Fold-cross for 10 and 50 Epochs

The more time we are training on Vanilla LSTM and Stacked LSTM, the more accuracy we are losing. However, Bidirectional LSTM is giving the opposite result. Accuracy is proportional to Training Time.

Algorithm	**Method**	**Epoch**	**Time**
LSTM	Test-Train	20	104 min
GRU	Test-Train	20	96 min
RNN	Test-Train	20	140 min

Table 5.4: Time comparison between algorithms based on Test-Train Split

Among three algorithms, RNN required more time for training, then LSTM. GRU needed less time.

Algorithm	Method	Epoch	Time
LSTM	Feature Selection	20	41 min
LSTM	Test-Train	20	94 min
LSTM	Fold Cross	20	121 min

Table 5.5: Runtime of LSTM for 20 Epochs

Fold Cross took more time for training than test-Train method. Feature Selection needed least time to train.

Method	LSTM	GRU	RNN
Test-Train	0.48	0.20	8.86
Fold Cross	0.27	0.27	0.51
Feature Selection	3.20	8.22	5.95

Table 5.6: Loss Comparison between Algorithms

Feature Selection lost more data and Fold Cross lost least data during training period.

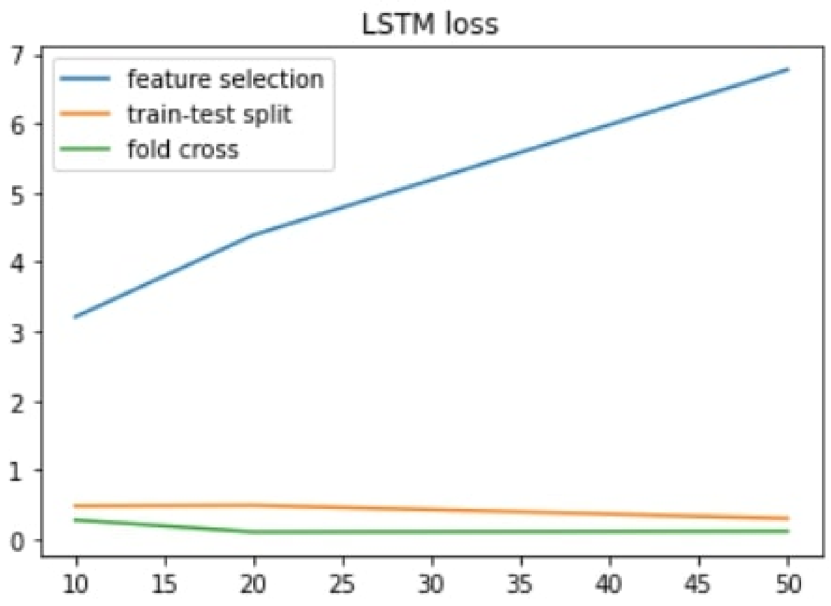


Figure 5.2: Data loss of LSTM during time

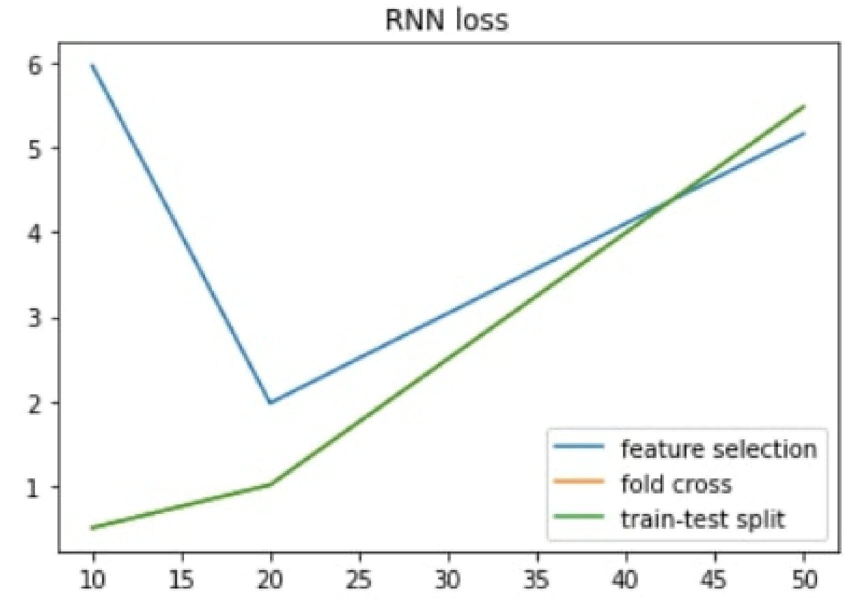


Figure 5.3: Data loss of RNN during time

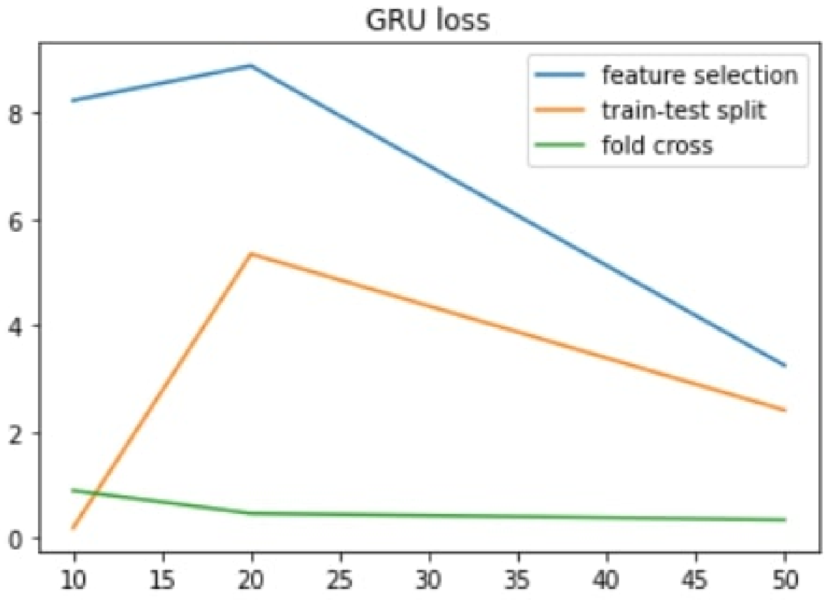


Figure 5.4: Data loss of GRU during time

5.2 Model accuracy on NSL-KDD dataset:

The Feature Selection approach is inapplicable to the NSL-KDD dataset since it lacks the massive amount of data found in the KDD dataset. Bidirectional LSTM, we feel, would also perform better on NSL-KDD based on our past work and data.

Method	**LSTM-Accuracy**	**GRU-Accuracy**	**RNN-Accuracy**
Fold Cross	0.79	0.71	0.81
Test-Train	0.61	0.31	0.71

Table 5.7: Accuracy comparison between Algorithms on NSL-KDD dataset

5.3 Previous Research

Algorithm	**Year**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
RNN-NSL [11]	2019	0.951	0.901	0.996	0.946
RNN-KDD	–	-	-	-	-
LSTM-NSL[11]	2019	0.964	0.931	0.990	0.959
LSTM-KDD [20]	2018	0.9885	0.9871	0.9867	0.100
GRU-NSL [11]	2019	0.939	0.885	0.987	0.933
GRU-KDD [20]	2018	0.9868	0.9877	0.9818	0.94
MLP-NSL [28]	2019	0.889	–	–	0.39
MLP-KDD [28]	–	0.805	–	–	0.194

Table 5.8: Different type approach of Deep learning Approach by using KDD-99 and NSL-KD dataset

Dataset	**Algorithm**	**Error Rate**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
ISCX2012[29]	LSTM	2.004%	97.996	98.108	97.887	97.997
Dd	GRU	3.208%	96.792	98.417	95.018	96.681

Table 5.9: Different type approach of Deep learning Approach by using ICX2012 dataset

After comparing all the analysis we have analysed that the Deep Learning model needs more huge data for training. Fold Cross method lost least data in training period and Feature Selection lost most data. RNN needed more time for training while GRU took less time. The Fold Cross method gave the best accuracy and Feature Selection did poor. Among all of the models, LSTM with the Fold Cross method performed best. It gives 98% accuracy. Then we used three types of LSTM. Vanilla LSTM gave 83% accuracy, Stacked LSTM 91% accuracy, and Fold Cross with Bidirectional LSTM gave 98% accuracy. We have found that using Fold Cross on combined GRU and Bidirectional LSTM gives 97% accuracy. NSL KDD didn't perform well because it required more pre-processing. Despite that, LSTM with Fold Cross gave better results compared to others which is 79%. We believe Bidirectional LSTM would do better on NSL KDD.

Chapter 6

Conclusion

To conclude, we would like to mention that we have experimented with a variety of algorithms and datasets for deep learning processes in IoT. The Internet of Things (IoT) is being gradually deployed to enable innovative applications. Various types of information are continually collected during the operation of these programs. Supervising and using IoT data to get insight into how to create a shrewd environment attracts both modern and academic ventures. While significant progress has been made in this area, there is still a necessity for advanced information expertise in IoT applications. If you are curious about our work, we obtained a very satisfying result in the KDD99 dataset by using the LSTM model with the fold cross technique performing the best and feature selection performing exceptionally poorly. Due to the fact that the fold cross approach lost the least data during the training process.On the other hand, feature selection discarded the most amount of data during data preprocessing. As a result, we obtain 98 percent accuracy using the LSTM fold cross. Additionally, we analyze deep learning using GRU and RNN. However, NSL KDD performed poorly since it requires further preprocessing, which we cannot give due to our device shortage. This was used to categorize the data, which aided in our accuracy. A critical component of the IoT framework is a capable information-insightful component capable of performing tasks such as grouping, prediction, relapse, and affiliation rule mining, among others. Removing substantial amounts of knowledge from raw IoT data is a very difficult task that is beyond the power of conventional information logical ideal models.DL is a good solution for a variety of arranging and forecasting tasks in the IoT since it can extract numerous levels of representations from the data sources. Additionally, DL is well-suited for demonstrating the multifaceted processes associated with diverse datasets. It consists of a variety of structures with a variety of applications. Time series are predicted using RNNs and LSTMs. Deep Learning has been widely used to decipher the data generated by IoT frameworks. Despite the fact that deep learning algorithms have been used in a variety of IoT applications, the invention must overcome a variety of obstacles in order to get the desired outcome. These difficulties are highlighted in this segment. To begin, we must acknowledge that our technical support for the world's current pandemic scenario presented a significant obstacle.Due to the fact that we are all students, none of us have advanced hardware devices for the purpose of LSTM training. We are aware that LSTM requires specialized devices and additional training time. We believe that if we can use our research lab to develop more advanced hardware devices, we will be able to provide more significant results than this one. Additionally, we face some time constraints. We attempted to use a model for LSTM and Gru in this work but could not do so due to a lack of training equipment and time. This task may require the usage of additional preprocessing procedures. For feature selection, we utilized pick kbest, which has a plethora of other possibilities but cannot focus on properly due to a lack of time. We may have utilized alternate ways in the stride of the k fold cross. Thus, these are the primary constraints on our task. We want to work on these areas in the near future. Deep learning methods for IoT intrusion detection have finally come in the future. This is only the beginning of the era of deep learning. There are numerous opportunities for work on it. It would be worthwhile to promote the development of models based on specified device kinds. As previously stated, deep learning is mostly used for feature selection. As a result, in the near future, it will develop some incredible functions for IDS. While deep learning models outperform classical AI techniques, their computationally complex nature precludes their usage in time-constrained IoT applications. As a result, approaches for reducing the computational complexity of these models without losing precision are required. Finally, it is clear to note that incorporating deep learning into IoT frameworks will expand the spectrum of usage of intrusion detection systems.

Bibliography

[1] J. Yuill, F. Wu, J. Settle, *et al.*, "Intrusion-detection for incident-response, using a military battlefield-intelligence process," *Computer Networks*, vol. 34, no. 4, pp. 671–697, 2000.

[2] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*, Citeseer, vol. 94, 2005, pp. 1723–1722.

[3] A. Graves, "Long short-term memory," in *Supervised sequence labelling with recurrent neural networks*, Springer, 2012, pp. 37–45.

[4] A. A. Aburomman and M. B. I. Reaz, "Review of ids development methods in machine learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 5, pp. 2432–2436, 2016.

[5] E. Hodo, X. Bellekens, A. Hamilton, *et al.*, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, 2016, pp. 1–6.

[6] A. Özgür and H. Erdem, "A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, e1954v1, 2016.

[7] T. Sherasiya and H. Upadhyay, "Intrusion detection system for internet of things," *Int. J. Adv. Res. Innov. Ideas Educ.(IJARIIE)*, vol. 2, no. 3, 2016.

[8] M. K. Putchala, "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)," 2017.

[9] J. Brownlee, "A gentle introduction to k-fold cross-validation," *Machine Learning Mastery*, vol. 2019, 2018.

[10] M. Nguyen, "Illustrated guide to lstm's and gru's: A step by step explanation," *Online] Towards Data Science*, 2018.

[11] R. Vinayakumar, K. Soman, P. Poornachandran, and S. Akarsh, "Application of deep learning architectures for cyber security," in *Cybersecurity and Secure Information Systems*, Springer, 2019, pp. 125–160.

[12] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the iot," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16 205–16 233, 2020.

[13] N. Chaabouni, "Intrusion detection and prevention for iot systems using machine learning," Ph.D. dissertation, Université de Bordeaux, 2020.

[14] M. S. Al-Daweri, K. A. Zainol Ariffin, S. Abdullah, *et al.*, "An analysis of the kdd99 and unsw-nb15 datasets for the intrusion detection system," *Symmetry*, vol. 12, no. 10, p. 1666, 2020.

[15] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[16] D. Learning, "Deep learning," *High-Dimensional Fuzzy Clustering*, 2020.

[17] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, e4150, 2021.

[18] N. Donges, *A guide to rnn: Understanding recurrent neural networks and lstm*, 2021.

[19] M. Maithem and G. A. Al-sultany, "Network intrusion detection system using deep neural networks," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1804, 2021, p. 012 138.

[20] *"recurrent neural network architectures toward intrusion detection" by wafaa anani*, https://ir.lib.uwo.ca/etd/5625/, (Accessed on 01/16/2022).

[21] *(pdf) on the importance of train-test split ratio of datasets in automatic landslide detection by supervised classification*, https://www.researchgate.net/publication/344299441_On_the_Importance_of_Train-Test_Split_Ratio_of_Datasets_in_Automatic_Landslide_Detection_by_Supervised_Classification?fbclid=IwAR2QmZAOFEnW K4EcpwM, (Accessed on 01/16/2022).

[22] *9.1. gated recurrent units (gru) — dive into deep learning 0.17.2 documentation*, https://d2l.ai/chapter_recurrent-modern/gru.html, (Accessed on 01/16/2022).

[23] *A 7 minute introduction to lstm. powerful deep learning algorithm widely... — by prateek karkare — ai graduate — medium*, https://medium.com/x8-the-ai-community/a-7-minute-introduction-to-lstm-5e1480e6f52a, (Accessed on 01/16/2022).

[24] *Deep learning approaches for intrusion detection in iiot networks – opportunities and future directions*, https://pdfs.semanticscholar.org/8656/22e027527be8270ec3c2169493 pdf, (Accessed on 01/16/2022).

[25] *Gated recurrent unit - what is it and how to learn*, https://analyticsindiamag.com/gated-recurrent-unit-what-is-it-and-how-to-learn/, (Accessed on 01/16/2022).

[26] *Gated recurrent unit architecture for context-aware recommendations with improved similarity measures -ksii transactions on internet and information systems (tiis) — korea science*, https://www.koreascience.or.kr/article/JAKO202011161036068.page, (Accessed on 01/16/2022).

[27] *Gated recurrent unit networks - geeksforgeeks*, https://www.geeksforgeeks.org/gated-recurrent-unit-networks/, (Accessed on 01/16/2022).

[28] *Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets - a review paper by rama devi ravipati, munther abualkibash :: Ssrn*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3428211, (Accessed on 01/16/2022).

[29] *Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets - a review paper by rama devi ravipati, munther abualkibash :: Ssrn*, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3428211, (Accessed on 01/16/2022).

[30] *Review of deep learning algorithms and architectures — ieee journals & magazine — ieee xplore*, https://ieeexplore.ieee.org/abstract/document/8694781?fbclid=IwAR2d2K8vBNVTH3ygXEsp4m46hH8ynHM-66jPhovkAxjh6-wJkag0PoarWOA, (Accessed on 01/16/2022).

[31] *The 9 most important applications of the internet of things (iot)*, https://www.fracttal.com/en/blog/the-9-most-important-applications-of-the-internet-of-things, (Accessed on 01/16/2022).

[32] T. Vaiyapuri, Z. Sbai, H. Alaskar, and N. A. Alaseem, "Deep learning approaches for intrusion detection in iiot networks–opportunities and future directions,"

[33] *What is an intrusion detection system? - palo alto networks*, https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids, (Accessed on 01/16/2022).

[34] *What is deep learning?* https://machinelearningmastery.com/what-is-deep-learning/, (Accessed on 01/16/2022).