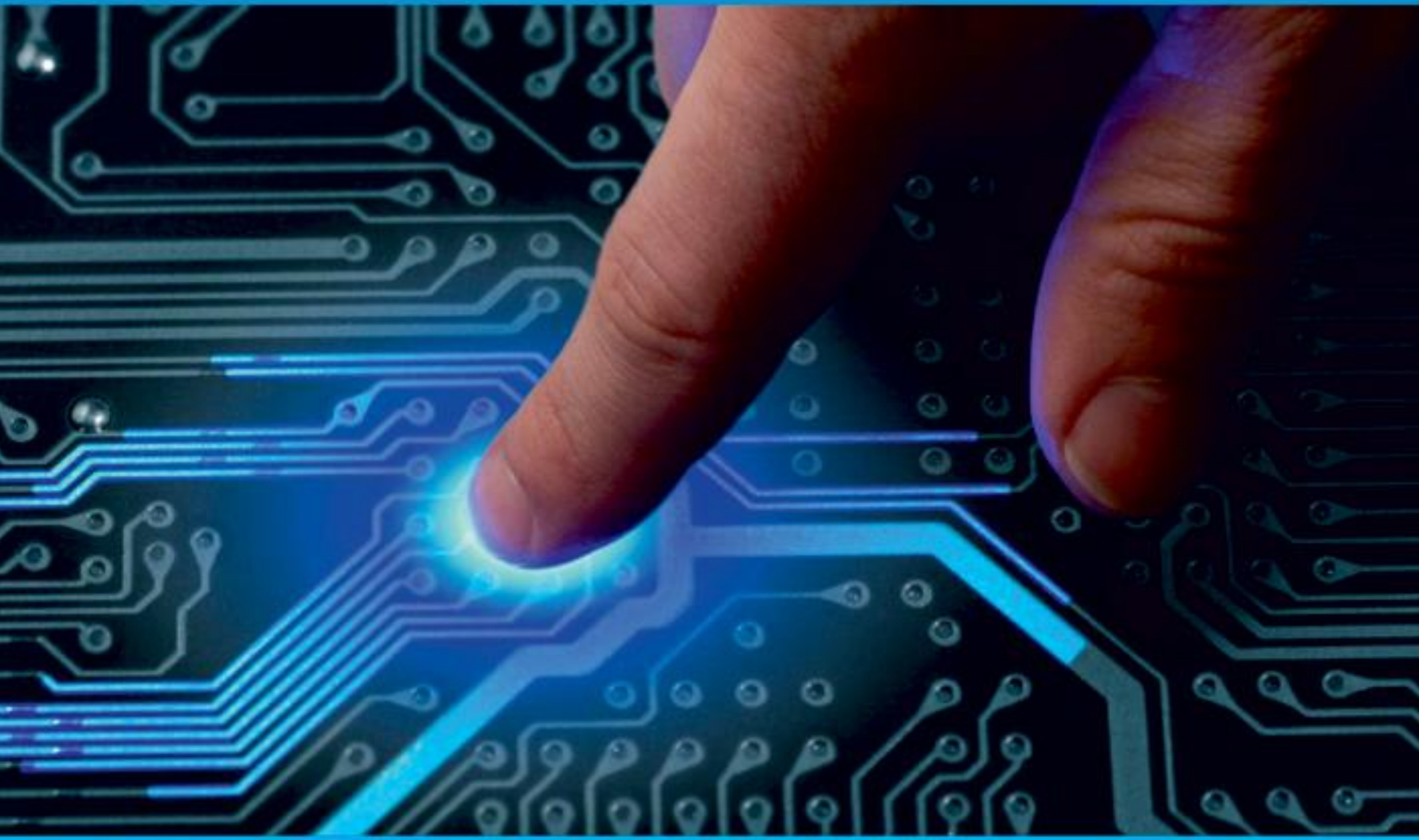
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 7, July 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 7, July 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1107057 |

Network Intrusion Detection using Recurrent Neural Network Algorithm

Bhagyasri B [1], Mr. Adarsh M J [2]

PG Student, Dept. of Master of Computer Applications, Jawaharlal Nehru New College of Engineering, Shivamogga, India

Assistant Professor, Dept. of Master of Computer Applications, Jawaharlal Nehru New College of Engineering, Shivamogga, India

ABSTRACT: Due to the large number of computer system faults and attackers' ingenuity, one of the most difficult difficulties network operators face recently is identifying network incidents of assault. We outline a deep learning strategy for intrusion detection systems to solve this issue. Introduce the suggested system's upgraded recurrent neural network (RNN), which employs intrusion detection to identify the kind of intrusion. Numerous studies come to the conclusion that in recent years, network infiltration has consistently increased, resulted in the theft of personal information, and become a key attack platform. Unauthorized action on a computer network is known as network intrusion. Consequently, it is necessary to create an efficient intrusion detection system. The results of the experiments conducted on the KDD-CUP'99 dataset demonstrate that, in terms of accuracy, detection rate, and false-positive rate, our technique significantly outperforms previous deep learning-based approaches.

KEYWORDS: Intrusion detection, Feature selection, linear correlation coefficient, deep learning, RNN

I. INTRODUCTION

To increase programming and network safety, a lot of effort has recently been put into creating detection systems for intrusions (IDSs). IDSs may be broadly categorized into two groups: misuse-based Intrusion Detection Systems(IDSs) and anomaly-based IDSs. Misuse-based Intrusion Detection Systems IDSs utilize a present signature to identify known attacks. As a result, dynamic characteristic updating is crucial, and IDS manufacturers routinely release new attack patterns. The misuse-based anIntrusion Detection Systems (IDS) for short however, is unable to account for the steadily increasing number of vulnerability and workarounds. Anomaly-based Intrusion Detection Systems (IDSs) are intended to record any departure from typical behaviour patterns. The final result of each automatic encoding in the current layer is utilized as the input of the autonomous encoder in the following layer in our Deep Auto-Encoder based Intrusion Detection System (DAE-IDS), which is made up of four auto-encoders. Moreover, when training the previous encoder is finished, training an auto-encoder begins. A soft maximal learner sorts the attack types from the input datasets in the final invisible layer. In order to do supervised fine-tuning and unsupervised feature learning, DAE-IDS performs intrusion detection. We do a number of exploratory tests to determine the ideal hyper-parameters since the performance of a deep model depends on the hyper-parameters used for model initiation. Additionally, we look into how hidden neurons and the quantity of hidden layers affect DAE-IDS performance. IDSs have received much investigation as defensive measures to detect undiscovered or zero-day threats. The regular network activity is modelled by anomaly-based IDSs, which then spot attempts as departures from that behaviour. As a result, an effective IDS can handle massive amounts of data with shifting patterns in real-time scenarios. Due to its enormous success in these disciplines, deep learning has now become a highly significant and popular research topic in the Machine Learning (ML) community. Deep learning is now the focus of research in order to overcome the aforementioned constraints. Deep learning, a sophisticated branch of machine learning, can get beyond some of the drawbacks of shallow learning. Deep learning is a cutting-edge machine learning approach that uses numerous information-processing layers in hierarchical designs to learn features or representations and identify patterns.

II. LITERATURE SURVEY

These days, the internet is a need and a part of daily life. The internet offers numerous benefits, but it has also given rise to many vices. Attacks have increased as a result of this. People as well as organizations may both be impacted by these assaults. As a result, research has long focused on how to make computer and network systems secure.Safeguarding data is a topic that is highly significant and must not be overlooked, according to all businesses that operate within the field of information technology.It is essential to fulfil the three fundamental requirements of integrity, accessibility, and secrecy, which form the foundation of every secure system.Attack detection is described as "the process of monitoring the events occurring in a network or computer environment and analyzing them for signs of hacking attempts, defined as attempts to damage the privacy, reliability, and availability, or to avoid the protective devices of a computer or network" by the National Institutes of Standards and Technology's website. [1],[2]. IDS can identify intrusive behaviours that jeopardize resource integrity, your availability, and confidentiality. IDSs may be used to identify many forms of illegal network activity and personal computer usage, but regular firewalls cannot.Based on the presumption that attackers behave differently than authorized users, intrusion detection is used [3].IDSs may generally be categorized into two categories: 1) ad hoc 2) depending on their detection methods, abuse (signature) detection [4].By examining the architecture of typical network traffic behaviour, the system identifies unknown or unexpected behaviour in network traffic in detection of anomalies. An incursion is defined as network traffic that deviates from a typical traffic pattern. Attack fingerprints are already pre-installed in the IDS for Exploitation (signature) detection. To find a network intrusion, an exact matching operation is done on the traffic against the installed signatures [5]. There will come a time when reliance on such procedures will result in inefficient and incorrect detection. The incorporation of machine learning and shallow learning approaches, such as naive Bayes, decision tree modelling, and support vector machine training (SVM), has been one of the key areas of concentration in IDS research in recent years [6]. The degree of precision of detection has increased because to the use of these approaches. However, these methods have disadvantages, such as the comparably high level of expert human contact required; data processing requires expert expertise. Similar to this, an enormous amount of training data is needed for operation (along with time overheads), which might be difficult in a complex and dynamic context [7]. Deep learning is now the focus of research in order to overcome the aforementioned constraints. Deep learning, a sophisticated branch of machine learning, can get beyond some of the drawbacks of shallow learning.Deep learning is a cutting-edge approach to machine learning that makes use of numerous information-processing planes in hierarchical structures to learn features or representations and categorize patterns [8].Because of its enormous success in these disciplines, deep learning has currently become a highly significant and productive research trend in the field of machine learning (ML) [9].The deep learning version for IDS operation inside contemporary networks is suggested in this study.The deep learning version for IDS operation inside contemporary networks is suggested in this study.

III. PROPOSED METHOD

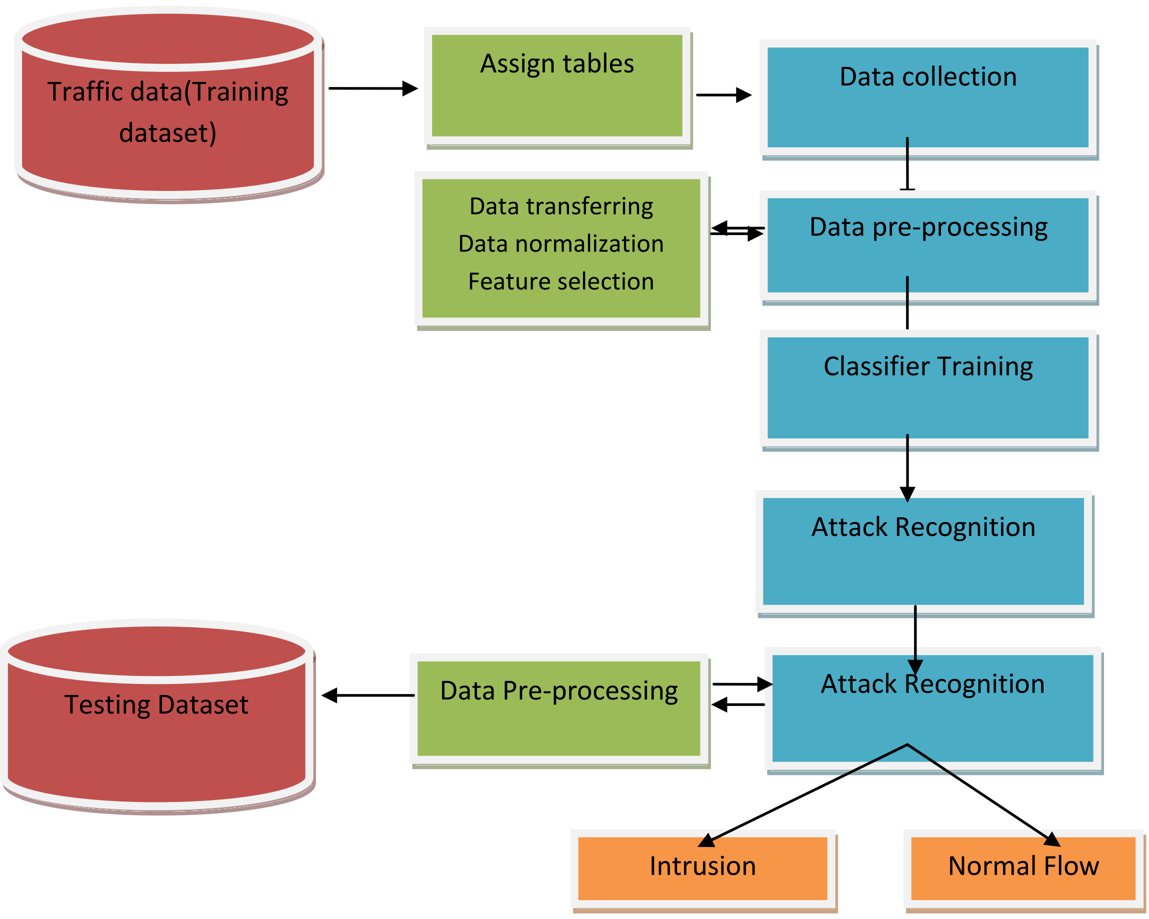


Fig. 1. Proposed System Architecture

Data Collection

Data collection is a kind of information assortment is the first and a basic move toward interruption recognition. The sort of information source and where information is gathered from are two determinate variables in the plan and the viability of an IDS.

Data Preprocessing

The information acquired during the period of information assortment are first handled to create the fundamental highlights, for example, the ones in KDD Cup 99 dataset. This stage contains three primary stages displayed as follows.

Data transferring

The trained classifier requires each record in the input data to be represented as a vector of real number. These symbolic features include the type of protocol (i.e., TCP, UDP and ICMP), service type (e.g., HTTP, FTP, Telnet and so on) and TCP status flag (e.g., SF, REJ and so on). The method simply replaces the values of the categorical attributes with numeric values

Data normalization

A fundamental stage of information preprocessing subsequent to moving all representative credits into mathematical qualities is standardization. Each element inside each record is standardized by the particular greatest worth and falls into a similar scope of [0-1]. The moving and standardization cycle will likewise be applied to test information. For KDD Cup 99 and to cause an examination with those frameworks that to have been assessed on various sorts of assaults we develop five classes. One of these classes contains simply the ordinary records and the other four hold various kinds of assaults (i.e., DoS, Test, U2R, R2L), individually.

Feature selection

Despite the fact that each association in a dataset is addressed by different highlights, not these elements are expected to construct an IDS. Consequently, it is essential to distinguish the most useful highlights of traffic information to accomplish better execution. In the past segment utilizing Calculation 1, an adaptable technique for the issue of component choice. Then, at that point, gradually the strategy adds highlights to the classifier individually. A ultimate choice of the ideal number of highlights in every technique is taken once the most elevated characterization precision in the preparation dataset is accomplished. The chose highlights for all datasets, where each column records the number and the files of the chose highlights as for the comparing highlight determination calculation. Moreover, for KDD Cup 99, the proposed include determination calculation is applied for the previously mentioned classes.

Algorithms Flow

Step 1: The data provided is sent to the computer network in one stage of time.
Step 2: Then, using the set of current input and the prior state, determine the present state of the system.
Step 3: For the following time step, the current moment in time becomes time1.
Step 4: Depending on the issue, one can travel back as many time steps and combine the data from all the prior stages.
Step 5: The final current state is used to determine the yield once all of the process steps have been finished.
Step 6: This mistake is then created once the output is checked against the goal output, which is the actual output.
Step 7: The network (RNN) is then trained by back-propagating the mistake to the network to update the weights afterward.

IV. IMPLEMENTATION

1. Input Dataset: Source Dataset NSL-KDD dataset is the input dataset. There are DoS, Probe, U2R, R2L, and Normal attacks in it. Since the NSL-KDD dataset was returned with unlabeled data, adding column headers to it was one of the first crucial steps. Information such as duration, protocol type, service, src bytes, dst bytes, flag, land, incorrect fragment, etc. are added to the headers of the total 41 columns.

   The following categories of assaults are listed:

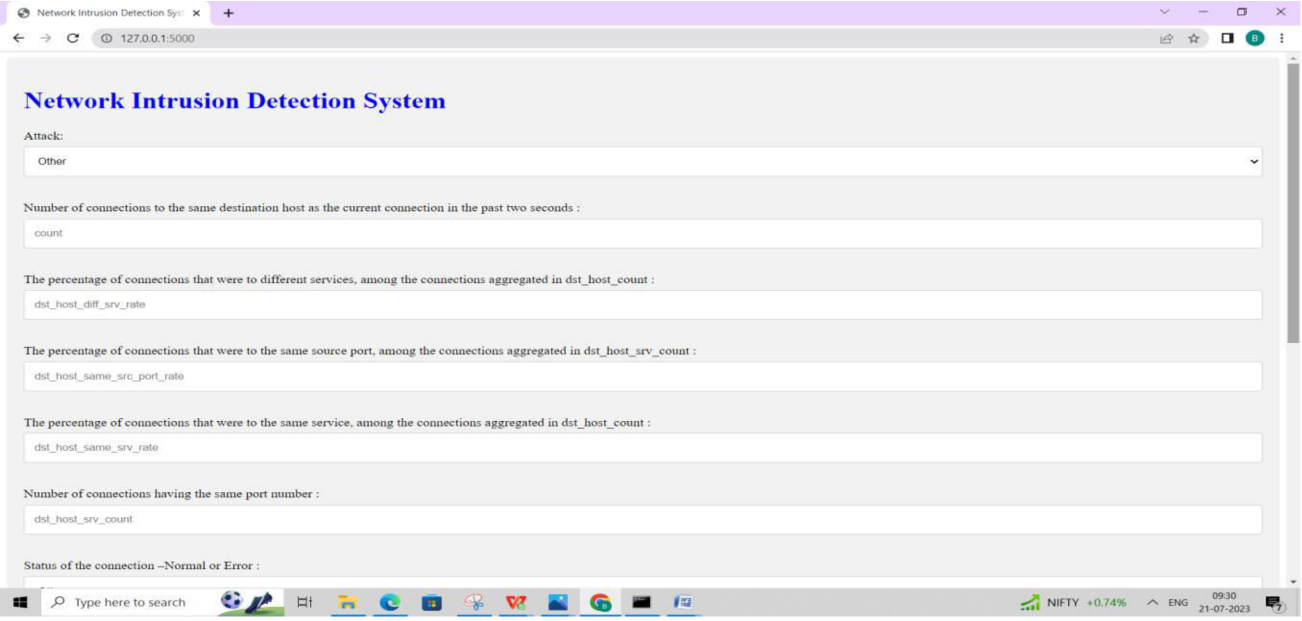
- Denial of Service (DoS): The attacker attempts to prevent genuine users from using a resource or system function by overloading it with erroneous requests. Denial of Service Attacks come in several varieties. By transmitting misshaped packets, some attacks attempt to take advantage of flaws in the protocol stack and network software.
- Probes: The probes don't harm anything on their own, but they supply useful information that may be utilized to mount an assault later. The attacker tries to look for services functioning on each system, legitimate IP addresses, or known vulnerabilities.
- Remote to user: The attacker has remote access to the system but not local access in a remote to user attack. To obtain access locally, the attacker tries to use a system weakness. Buffer overflows in network server software as well as improperly and inadequately designed systems are among the vulnerabilities.
- User to root: The attacker has local access to the system from user to root. To get access to the super-user account, the hacker tries to utilize a system defect. Overflows in the buffer are a frequent vulnerability, including other flaws in temporary handling of files and race situations.

2. Data-Preprocessing: Pre-processing of Data To make the system more effective, the data has to be pre-processed. To prevent several problems, such as a high detection rate ratio, false alarms, and instruction overhead, the raw data is processed before input rather than being given directly.

3. Classifier Training: Classifier Education The ideal selection of features is chosen, and this subset is subsequently used in the LS-SVM classifier training phase. Five LS-SVM classifiers must be used since SVMs can only handle binary classification issues and the NSL KDD Dataset has five optimum feature subsets chosen for each class.
4. Attack Recognition: Attack Identification The most associated and significant characteristics are used to train the classifier, which may be used to distinguish between normal and incursion traffic using the learned classifier that has been preserved. The trained model is then used to detect intrusions using the test data. Normal data are records that match the normal class, while attacks are records that don't match the usual class.

V. RESULTS

Network Intrusion Detection System



Network Intrusion Detection System

Attack:

Other

Number of connections to the same destination host as the current connection in the past two seconds :

count

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

dst_host_diff_srv_rate

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

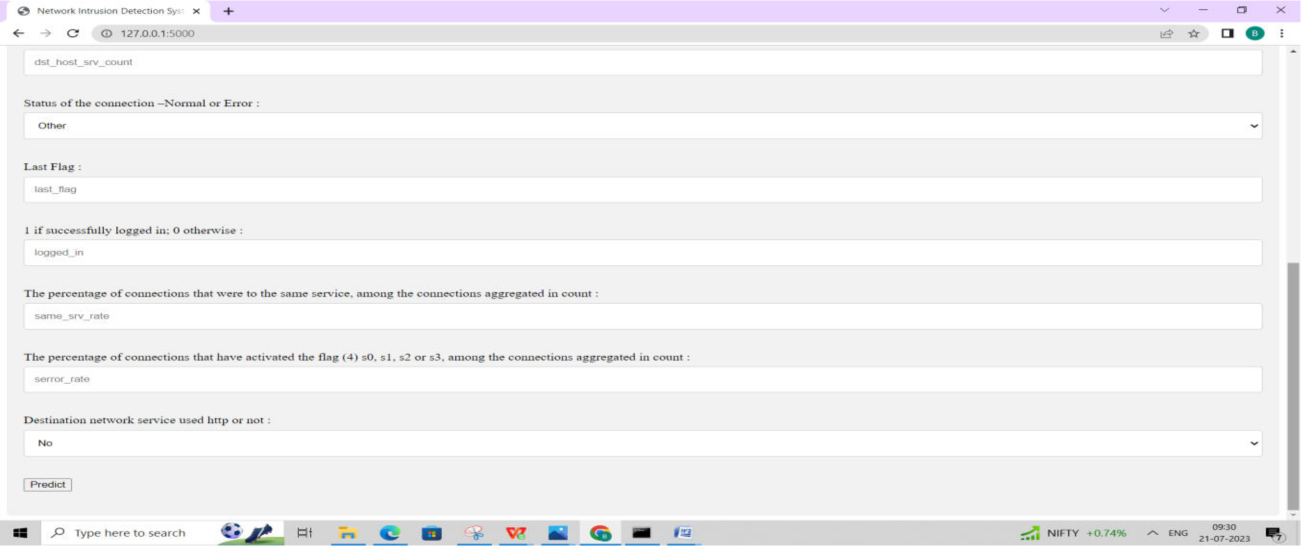
dst_host_same_srv_rate

Number of connections having the same port number :

dst_host_srv_count

Status of the connection –Normal or Error :

Fig 2. For first front design for user input for prediction of attack



dst_host_srv_count

Status of the connection –Normal or Error :

Other

Last Flag :

last_flag

1 if successfully logged in; 0 otherwise :

logged_in

The percentage of connections that were to the same service, among the connections aggregated in count :

same_srv_rate

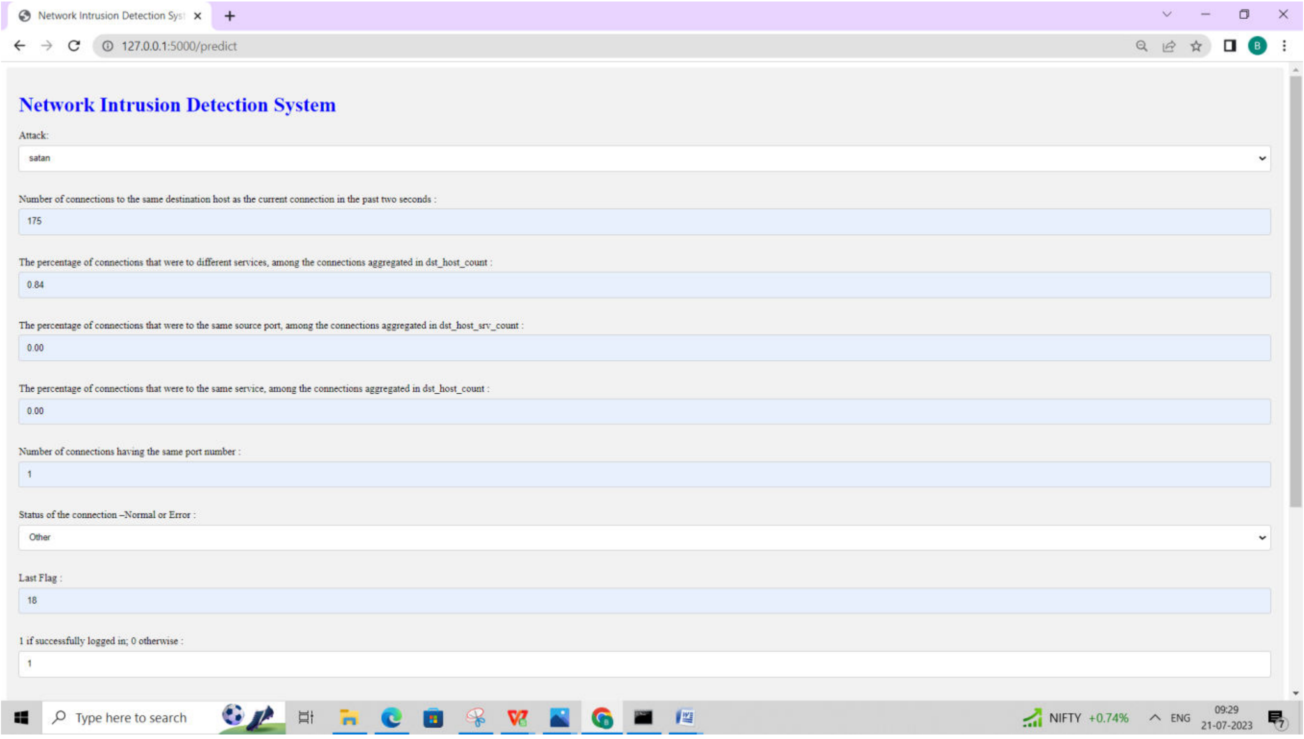
The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

serror_rate

Destination network service used http or not :

No

Predict



Network Intrusion Detection System

Attack:

satan

Number of connections to the same destination host as the current connection in the past two seconds :

175

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

0.84

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

0.00

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

0.00

Number of connections having the same port number :

1

Status of the connection –Normal or Error :

Other

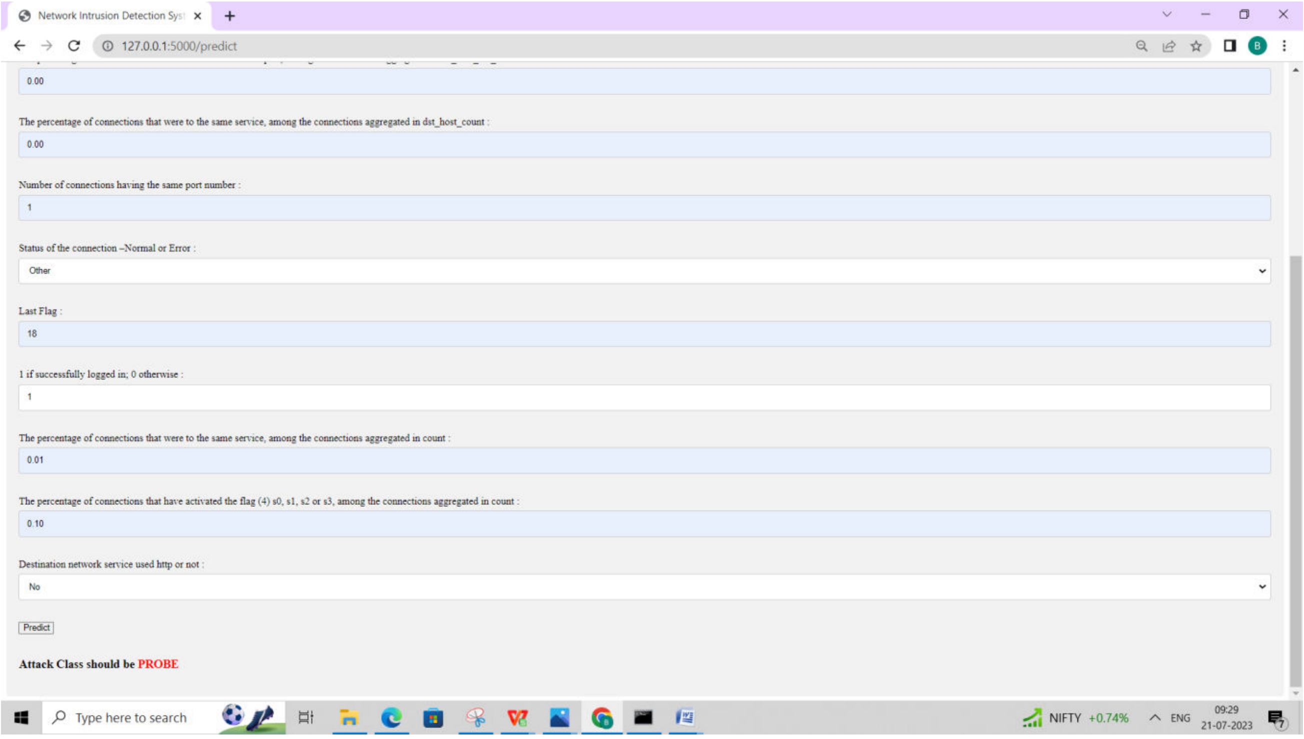
Last Flag :

18

1 if successfully logged in; 0 otherwise :

1

Fig3: for second after giving input the predicted output will be displayed



0.00

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

0.00

Number of connections having the same port number :

1

Status of the connection –Normal or Error :

Other

Last Flag :

18

1 if successfully logged in; 0 otherwise :

1

The percentage of connections that were to the same service, among the connections aggregated in count :

0.01

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

0.10

Destination network service used http or not :

No

Predict

Attack Class should be PROBE

VI. CONCLUSION

The development of effective Intrusion Detection Systems (IDSs) has become increasingly popular as a result of the dramatic rise in network traffic and various types of assaults. In this article, we discussed a technique using a deep auto-encoder to enhance the intrusion detection system. We have outlined a deep learning strategy for intrusion detection in this research. Applications for intrusion detection are examined for a few popular deep learning architectures. In the context of deep learning, the auto-encoder is one of the most fascinating algorithms to extract features from the high-dimensional data. This led to the suggestion of the feature learning method. Then, building on this, a novel classification model using the Recurrent Neural Network Classification Algorithm was proposed. The outcome demonstrates that the suggested technique provides excellent levels of precision, recall, and accuracy while requiring less training time. Utilizing a recurrent neural network, the suggested NIDS system's accuracy is only increased by 8%.

REFERENCES

1. N. Gao, L. Gao, Q. Gao, and H. Wang, “An intrusion detection model based on deep belief networks,” in Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, 2014, pp. 247– 252.

2. N. Gao, L. Gao, Q. Gao, and H. Wang, “An intrusion detection model based on deep belief networks,” in Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, 2014, pp. 247– 252.
3. K. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” in Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on, 2016, pp. 195–200.
4. K. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” in Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on, 2016, pp. 195–200.
5. J. Kim, J. Kim, H. L. T. Thu, and H. Kim, “Long short-term memory recurrent neural network classifier for intrusion detection,” in Platform Technology and Service (Platicon),2016 International Conference on, 2016, pp. 1–5.
6. Adarsh M, J and Sumanth M “Effective Heart Disease Diagnosis Using Machine Learning Techniques”, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume – 2, Issue – 6, June 2022.
7. Adarsh M, J and Pallavi S N “Road Traffic Congestion using Local Binary Pattern “, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume – 2, Issue – 6, June 2022
8. Adarsh M, J and Madhura S M “Recommendations for Agricultural Crops Based on Productivity and Season”, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume – 1, Issue – 9, June 2022.
9. Adarsh M, J and Chandana K C “Python Based Naive Bayes Classifier for Spam Comment Detection” International Journal of Combined Research & Development (IJCRD) eISSN:2321-225X; pISSN:2321-2241 Volume: 11; Issue: 6; June -2022
10. Adarsh M J, Md. Irshad Hussain B,” A Survey on Digitization of Historical Document with Image Enhancement Techniques” - Jan. 18 | Volume XII | Issue I | Article Number: 14 | ISSN 2321-3469. In International Journal of Computer Engineering and Applications.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING