Abstract:

A Study on the Efficacy of Machine Learning Models in Intrusion Detection Systems

Praveen Kumar Shukla¹, Dr C S Raghuvanshi², Dr Hari Om Sharan³

¹Department of computer science & Engineering, Rama University, Kanpur 209217, INDIA

Email Id: praveenshukla26@gmail.com

² Department of computer science & Engineering, Rama University, Kanpur 209217, INDIA

Email Id: drcsraghuvanshi@gmail.com

³ Department of computer science & Engineering, Rama University, Kanpur 209217, INDIA

Email Id: deanengineering@ramauniversity.ac.in

Article History: Received: 30-03-2024 *Revised:* 20-05-2024 *Accepted:* 01-06-2024

The electronics industry has seen a rise in demand for faster and more affordable delivery due to developments in information technology. Technology is developing quickly, which simplifies living but also presents a number of security issues. As the Internet has grown over time, so too have the amount of online attacks. The intrusion detection system (IDS) is one of the supporting layers that can be utilized for information security. IDS avoids questionable network activity and provides a pristine environment for conducting business. In the process of building an e-commerce system, the most challenging aspect is ensuring user security during online transactions. Security methods for intrusion detection were investigated in this study. The need for ongoing intrusion detection monitoring stems from the need for continued technological adaptation, which leads to a comparison of adaptive artificial intelligence-based intrusion detection systems. This paper demonstrates the use of reinforcement learning (RL) and regression learning-based intrusion detection systems (IDS) to very challenging problems, including resource allocation and input feature selection.

Keywords: reinforcement learning, linear regression, logistic regression, intrusion detection, KDD dataset.

1. INTRODUCTION

Electronic commerce (E-commerce) is the only way to meet the needs of business organizations in the modern, digital age. Delivery times are accelerated by e-commerce, which provides premium products at reduced costs. Sometimes it's referred to as paperless business information sharing. Due to its many features, including non-cash payments, round-the-clock services, improved marketing management, pre- and post-sale services, and prompt and trustworthy customer communication, business partners prefer it. It acts as a single platform for information exchange, maintaining strategies, and assisting with international company operations. Apart from these advantages, e-commerce has two main non-technical and technological drawbacks. The main technological shortcoming of the system is its security, which is the result of poor algorithm implementation. Providing security for online transactions on the user's end is a difficult task. The owner and user will suffer an unsupportable financial loss. The reason that causes this loss in e-commerce is the

downtime brought on by an intrusion attack. As a result, this study uses intrusion detection algorithms to analyse security algorithms.

2. LITERATURE REVIEW

PAPER TITLE	RESEARCH TECHNIQUE	FUTURE SCOPE
Semi-supervised multi- layered clustering model for intrusion detection. Digital Communications and Networks 4(4): 277-286.	In this paper SMLC (an ensemble model of multiple randomized layers using K- means algorithm) is made and then this model result is compared with the known ML algorithms such as random forest, adaboost, bagging to detect the accuracy of the model. The SMLC model reduces the computational cost and can process large amounts of data.	SMLC has a relatively high testing time. In future work should be done to improve the scalability of SMLC, automation of its parameter tuning process and reduction of its testing time.
Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection- Procedia Computer Science 89(1): 117- 123.	In this paper four ML algorithms are used on a NSL-KDDdataset and their accuracy, F-scores, precision, recalls are compared and out of the four models Random forest model has given the best results.	In future work can be extended by considering the classifiers for multiclass classification and considering only the important attributes for the intrusion detection
IDS for IoT-based smart environments: a survey Journal of Cloud Computing 7 (1): 21	In this paper the IDS system is integrated with the IOT smart applications to prevent attacks and security issues. Here the PCA (Principal component analysis) algorithm is used in IDS as it is a lightweight algorithm for IOT- based environment and descriptive multivariate method for handling quantitative data and can be extended to deal with mixed measurement level data.	In future work can be done in the areas to reduce the monitored traffic and increase the processing time and automate the detection process without human intervention.
Intrusion detection in smart cities using Restricted Boltzmann Machines." Journal of Network and Computer Applications 135 (1): 76-83.	In this paper two main methods are used. One is the RBM(Restricted Boltzmann model) for feature extraction and the other is classifier model for attack detection.	In future work can be done to reduce the error percentage and detect variety of attacks.

3. MATERIALS AND METHODOLOGY

Step1: Import libraries and load dataset

+ Code + Text IMPORTING LIBRARIES	<pre>+ Code + Text</pre>		Edit	View Ins	ert Runt	time Tools	s Help	All cha	ng <u>es sa</u>	ved							E	Com	iment	** \$	Share
<pre>IMPORTING LIBRARIES</pre>	<pre>MPORTING LIBRARIES * importing required libraries import numpy as np import pandas as pd import seaborn as sns import matpletile.pyplot as plt import pickle from sklearn.preprocessing import MinMaxScaler from sklearn.preprocessing import LabelEncoder from sklearn.import metrics from sklearn.model_selection import train_test_split from sklearn.model_selectin test_split from sklearn.model_select</pre>	+ Code	+	Text													RAM Disk		•	🖍 Edi	iting
 	<pre> # importing required libraries import numpy as np import pandas as pd import seaborn as sns import matplotlib.pyplot as plt import pickle from sklearn.preprocessing import MinMaxScaler from sklearn.preprocessing import StandardScaler from sklearn.preprocessing from sklearn.metrics import tabelEncoder from sklearn.metrics import classification resource</pre>	IMPO	RTI	NG LIBRAF	RIES																
<pre>from os import path from sklearn.preprocessing import MinMaxScaler from sklearn.preprocessing import StandardScaler from sklearn.metrics from sklearn import metrics from sklearn import preprocessing from sklearn.metrics import accuracy_score from sklearn.metrics import classification report</pre>	<pre>from os import path from sklearn.preprocessing import MinMaxScaler from sklearn.preprocessing import StandardScaler from sklearn.import metrics from sklearn.import metrics from sklearn.metrics import accuracy_score from sklearn.metrics import accuracy_score from sklearn.metrics import train_test_split from sklearn.metrics import classification report</pre>		# im impo impo impo impo	mporting re- ort numpy a ort pandas ort seaborr ort matplot	equired as np as pd n as sns tlib.pyp	libraries ; ;	lt											<u>+</u> <u>+</u>			
 ✓ Os completed at 10:21 AM ▲ ↓ ⊕ 目 ↓ □ = 	<pre>> 0s completed at 10:21 AM</pre>	f f	from from	n sklearn.p n sklearn.p n sklearn.p	preproce	essing imp essing imp essing imp	oort Min oort Sta oort Lab	nMaxSca andardS belEnco	ler caler der												
	↑ ↓ ⊕ □ ↓ □ : data = pd.read_csv('/content/UNSM_NBI5.csv') [3] data.head(n=5)	t t t	from from from	n sklearn : n sklearn : n sklearn.m n sklearn.m	import m import p netrics nodel_se	import ac election i	ing curacy mport	_score train_t	est_spl	it											
	<pre>(3) data.head(n=5)</pre>	+ + + +	from from from from	n sklearn : n sklearn : n sklearn.n n sklearn.n	import m import p netrics nodel_se	metrics preprocess import ac election i import cl	ing curacy mport 1 assifi	_score train_t	est_spl	it ✓ Os	compl	eted at 10:21	AM						-		
id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_ftp_cmd ct_fl		4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	from from from from from from from from	sklearn: n sklearn. n sklearn. n sklearn. n sklearn. head(n=5) d dur	<pre>import metrics metrics csv('/cc proto</pre>	service	sing curacy_ mport 1 assifi SW_NB15 SW_NB15	_score train_t cation .csv') spkts	est_spl report	it ✓ Os sbytes	compl	eted at 10:21	AM	ct_dst_sport_1	tm ct_d	st_src_ltm	↑ is_ftp_i	√ es	ct_ftp.	* [_ _cmd	i : ct_flu
id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_ftp_cmd ct_fl 0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0	0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0	da	from from from from from from from from	n sklearn : n sklearn n n sklearn n n sklearn n n sklearn n head(n=5) d dur 1 0.121478	<pre>import m import p netrics nodel_se netrics csv('/cc proto tcp</pre>	errics preprocess import ac election i import cl ontent/UNS service	sing mport t assifi SW_NB15 SW_NB15	_score train_t .csv') spkts 6	est_spl report dpkts 4	it ✓ Os sbytes 258	compl dbytes 172	eted at 10:21 rate 74.087490	AM	ct_dst_sport_1	tm ct_d	st_src_ltm 1	↑ is_ftp_:	v es login 0	> ■ ¢	≭ ⊾ _cmd	∎ : ct_flv
id dur proto service state spkts dpkts sbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_ftp_cmd ct_fl 0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0 1 2 0.649902 tcp - FIN 14 38 734 42014 78.473372 1 2 0 0	0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0 1 2 0.649902 tcp - FIN 14 38 734 42014 78.473372 1 2 0 0	da (3) da	from from from from from from from from	<pre>sklearn : sklearn : s</pre>	<pre>import m import p metrics nodel_se metrics csv('/cc proto tcp tcp</pre>	errics import ac election i import cl ontent/UNS service -	sing couracy_ mport t assifie SW_NB15 SW_NB15 SW_NB15 FIN	_score train_t _cstion _ 	est_spl renort dpkts 4 38	it ✓ 0s sbytes 258 734	compl dbytes 172 42014	eted at 10:21 rate 74.087490 78.473372	AM	ct_dst_sport_1	tm ct_d 1	st_src_ltm 1 2	↑ is_ftp_1	↓ œ login 0 0	ct_ftp	≭ [.] _cmd 0 0	i : ct_flw
id dur proto service state spkts dpkts sbytes rate ct_dst_sport_1tm ct_dst_src_1tm is_ftp_login ct_ftp_cmd ct_f	0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0 1 2 0.649902 tcp - FIN 14 38 734 42014 78.473372 1 2 0 0 2 3 1.623129 tcp - FIN 8 16 364 13186 14.170161 1 3 0 0	da (3) da 0 1	from from from from from from from from	<pre>sklearn : sklearn : s</pre>	mport m import p metrics csv('/cc proto tcp tcp tcp	errics import as election i immort cl ontent/UNS service	sing couracy_ mport t assifi SW_NB15 SW_NB15 SW_NB15 SW_NB15 FIN FIN	_score train_t ration .csv') spkts 6 14 8	est_spl cenort dpkts 4 38 16	it ✓ 0s sbytes 258 734 364	compl dbytes 172 42014 13186	rate 74.087490 78.473372 14.170161	AM	ct_dst_sport_1	tm ct_d 1 1	st_src_ltm 1 2 3	↑ is_ftp_:	↓ c= login 0 0 0	ct_ftp	≭ ⊾ _cmd 0 0 0	i : ct_flw
id dur proto service state spkts dpkts sbytes rate ct_dst_sport_1tm ct_dst_src_1tm is_ftp_login ct_ftp_cmd ct_ftp_cmd	0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0 1 2 0.649902 tcp - FIN 14 38 734 42014 78.473372 1 2 0 0 2 3 1.623129 tcp - FIN 8 16 364 13186 14.170161 1 3 0 0 3 4 1.681642 tcp ftp FIN 12 12 628 770 13.677108 1 3 1 1	da (3) da 0 1 2 3	from from from from from from from from	<pre>sklearn : sklearn : s</pre>	proto tcp tcp tcp	etrics import ac election i immort cl ontent/UNS service - - - -	state FIN FIN FIN FIN	_score train_t cation .csv') spkts 6 14 8 12	est_spl renort dpkts 4 38 16 12	it	compl dbytes 172 42014 13186 770	rate 74.087490 78.473372 14.170161 13.677108	AM	ct_dst_sport_1	tm ct_d 1 1 1	st_src_ltm 1 2 3 3	↑ is_ftp_:	↓ c=2 login 0 0 1	ct_ftp,	cmd 0 0 1	∎ : ct_flw
id dur proto service state spkts dpkts sbytes rate ct_dst_sport_1tm ct_dst_src_1tm is_ftp_login ct_ftp_cmd ct_ftp_cmd	0 1 0.121478 tcp - FIN 6 4 258 172 74.087490 1 1 0 0 1 2 0.649902 tcp - FIN 14 38 734 42014 78.473372 1 2 0 0 2 3 1.623129 tcp - FIN 8 16 364 13186 14.170161 1 3 0 0 3 4 1.681642 tcp ftp FIN 12 12 628 770 13.677108 1 3 1 1 4 5 0.449454 tcp - FIN 10 6 534 268 33.373826 1 40 0 0	da (3) da (3) da (3) da (4) (4) (4) (4) (4) (4) (4) (4) (4) (4)	from from from from from ita. ita. ita. ita. ita. ita. ita. ita.	sklearn : sklearn : sklearn : sklearn : sklearn r sklearn r sklearn r sklearn r sklearn r sklearn r sklearn r sklearn r sklearn r sklearn s sklearn s sklear	proto tcp tcp tcp tcp tcp	service	state FIN FIN FIN FIN FIN FIN	_score train_t cation .csv') spkts 6 14 8 12 10	dpkts 4 38 16 12 6	it ✓ 0s sbytes 258 734 364 628 534	compl dbytes 172 42014 13186 770 268	rate 74.087490 78.473372 14.170161 13.677108 33.373826	AM .	ct_dst_sport_1	tm ct_d 1 1 1 1 1	st_src_ltm 1 2 3 3 40	↑ is_ftp_i	↓ c= login 0 0 1 0	ct_ftp.	≭ □ _cmd 0 0 1 1 0	∎ : ct_flw

Step2: Data pre-processing

Scaling numerical Attributes: Taking away the mean and scaling to one variance will normalize the characteristics. Statistics regarding the cases in the training phase are independently targeted and adjusted by computing each attribute. The data that follows are then modified using the standard error and average that were created earlier. Many approximations require dataset uniformity; if the specific qualities do not match standard data that is normally distributed, they may perform poorly. Pre-processing of Data

- Dataset had **45 attributes** and **175341 rows**.
- The dataset has 81173 rows and 45 attributes after the null values were removed.
- Utilizing given data type information from the features dataset, the data type of attributes is converted.

		D		A 11 - 1									Comment	Share	-
File Edit Vie	ew Insert	Runtime To	ools Hel	p <u>All chai</u>	ng <u>es save</u>	ed							RAM	A	
+ Code + Tex	xt											~	Disk 🔳 🔍 👻	/ Editing	· ^
[4] data.in	10()														
<pre><class #="" 0="" 1="" 10="" 11="" 12="" 13="" 2="" 3="" 4="" 5="" 6="" 7="" 8="" 9="" <="" col="" data="" dbj="" dlu="" dpi="" dtt="" duu="" id="" pre="" pro="" rai="" rangein="" sbj="" set="" slu="" spi="" sta="" str=""></class></pre>	'pandas.co idex: 17534)lumns (tot)lumn i r voto rrvice tate okts okts okts okts okts okts okts okts	pre.frame.D 11 entries, 13 45 colu Non 175 175 175 175 175 175 175 175	ataFrame 0 to 17 mns): -Null Co 	'> 5340 unt Dty null int null obj null obj null obj null int null int null int null int null int null int null fla null fla	/pe 64 ect ject 64 64 64 64 64 64 64 64 64 64 64 64 64										
13 dla 14 sla	load Loss	175	341 non- 341 non-	null flo null int	at64 :64										
15 dla	oss	175	341 non-	null int	:64										
						• 03	comp	eteu at Te	.21 AM						• ^
	SIONDEL	IEC HOIN 3	5131EIVI	.ipynb	24										
File Edit V	View Insert	t Runtime	Tools H	Help <u>All c</u>	changes	saved								omment	Share
File Edit V + Code + Te	View Insert	t Runtime	Tools H	Help <u>All o</u>	int64	saved							RAM I Disk I		Share Editing
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes memory	View Inserf ext tt_src_ltm tt_srv_dst is_sm_ips_; nttack_cat label : float64(/ usage: 66	t Runtime 1 ports 1 1 (11), int6 0.2+ MB	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o	n-null n-null n-null n-null n-null n-null bject(4)	int64 int64 int64 object int64	saved							RAM Disk	mment	* Editing
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes memory [5] data[da	View Inserf ext :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float64(r usage: 60 fata['serv:	t Runtime 1 ports 1 (11), int6 0.2+ MB ice']=='-'	Tools H 75341 na 75341 na 75341 na 75341 na 4(30), a	n-null nn-null nn-null nn-null nn-null bject(4)	int64 int64 int64 object int64	saved							RAM Disk		Share Editing
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes: memory [5] data[di	View Inseri 'ext :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float64(/ usage: 66 lata['serv: id	t Runtime 1 ports 1 1 (11), int6 0.2+ MB ice']=='-' dur	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o] proto	n-null m-null m-null m-null m-null bject(4)	int64 int64 int64 object int64 state	spkts	dpkts	sbytes	dbytes	rate		ct_dst_sport_ltm	RAM Disk	is_ftp_lo	Editing
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes: memory [5] data[di	View Inseri fext :t_src_ltm :t_srv_dst is_sm_ips_f sttack_cat label ::float64(r usage: 66 data['serv: id 1	t Runtime 1 ports 1 1 (11), int6 0.2+ MB 1 1 (11), int6 0.2+ MB 1 0.121478	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o] proto tcp	n-null n-null n-null n-null n-null bbject(4) service	int64 int64 int64 object int64 state FIN	spkts 6	dpkts 4	sbytes 258	dbytes 172	rate 74.087490	••••	ct_dst_sport_ltm 1	RAM Disk Control of the second	is_ftp_lo	<pre>Share Share State Gin ct_ft 0</pre>
File Edit V + Code + Te 40 cr 41 cr 42 is 43 at 44 la dtypes: memory [5] data[da 0 1	View Inserf fext :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float64; r usage: 66 data['serv: id 1 2	t Runtime 1 ports 1 (11), int6- 0.2+ MB 1 1 (11), int6- 0.2+ MB 1 0.121478 0.649902	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o] proto tcp tcp	n-null m-null m-null m-null m-null bject(4) service	int64 int64 int64 object int64 state FIN FIN	spkts 6 14	dpkts 4	sbytes 258 734	dbytes 172 42014	rate 74.087490 78.473372	••••	<pre>ct_dst_sport_ltm 1 1</pre>	RAM Disk Ct_dst_src_ltm 1	is_ftp_lo	Share California Ct_ft O O
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes: memory [5] data[d: 0 1 2	View Inseri Text :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float644 / usage: 64 Jata['serv: id 1 2 3	t Runtime 1 ports 1 (11), int6 0.2+ MB (ice']=='-' dur 0.121478 0.649902 1.623129	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o] proto tcp tcp tcp	m-null m-null m-null m-null m-null bject(4) service	int64 int64 int64 object int64 state FIN FIN FIN	spkts 6 14 8	dpkts 4 38 16	sbytes 258 734 364	dbytes 172 42014 13186	rate 74.087490 78.473372 14.170161	••••	<pre>ct_dst_sport_ltm</pre>	RAM Disk Control of the second	is_ftp_lo	Share Cliting
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes: memory [5] data[di 0 1 2 4	View Inseri Text it_src_ltm it_srv_dst is_sm_ips_; sttack_cat label s: float64; / usage: 6; data['serv: id 1 2 3 5	t Runtime	Tools H 75341 no 75341 no 75341 no 75341 no 75341 no 4(30), o]] proto tcp tcp tcp tcp	n-null n-null n-null n-null n-null bject(4) service	int64 int64 int64 object int64 state FIN FIN FIN FIN	saved spkts 6 14 8 10	dpkts 4 38 16 6	sbytes 258 734 364 534	dbytes 172 42014 13186 268	rate 74.087490 78.473372 14.170161 33.373826		<pre>ct_dst_sport_ltm 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</pre>	RAM Disk Disk Ct_dst_src_ltm 1 2 3 40	is_ftp_lo	Share Celting
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes memory [5] data[da 0 1 2 4	View Inserf Text :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float644 / usage: 64 data['serv: id 1 2 3 5	t Runtime 1 ports 1 (11), int6 0.2+ MB ice']=='-' dur 0.121478 0.649902 1.623129 0.449454 0.280527	Tools H 75341 no 75341 no 7534	n-null n-null n-null n-null n-null bject(4) service	state FIN FIN FIN	saved spkts 6 14 8 10	dpkts 4 38 16 6	sbytes 258 734 364 534	dbytes 172 42014 13186 268	rate 74.087490 78.473372 14.170161 33.373826	••••	ct_dst_sport_ltm 1 1 1 1 1	Ct_dst_src_ltm 1 2 3 40	is_ftp_lo	share Editing diting ct_ft 0 0 0 0 0 0 0
File Edit V + Code + Te 40 ct 41 ct 42 is 43 at 44 la dtypes: memory [5] data[di 0 1 2 4 5	View Inseri Text :t_src_ltm :t_srv_dst is_sm_ips_; attack_cat label :: float644 / usage: 64 jata['serv: id 1 2 3 5 6	t Runtime 1 ports 1 (11), int6 0.2+ MB (ice']=='-' dur 0.121478 0.649902 1.623129 0.449454 0.380537	Tools H 75341 no 75341 no 7534	n-null m-null m-null m-null m-null bject(4) service	state FIN FIN FIN FIN FIN	saved spkts 6 14 8 10 10	dpkts 4 38 16 6 6	sbytes 258 734 364 534	dbytes 172 42014 13186 268	rate 74.087490 78.473372 14.170161 33.373826 39.417980	••••	<pre>ct_dst_sport_ltm</pre>	RAM □ Disk □ ct_dst_src_ltm 1 1 2 3 40 40 40	is_ftp_lo	Share Cliting
File Edit V + Code + Te 40 ct 41 ct 42 i: 43 at 44 la dtypes: memory [5] data[di 0 1 2 4 5 	View Inseri Text tt_src_ltm tt_srv_dst is_sm_ips_; attack_cat label s: float64(/ usage: 6(data['serv: id 1 2 3 5 6 	t Runtime 1 ports 1 (11), int6- 0.2+ MB 1 (11), int6- 0.2+ MB 1 (11), int6- 0.2+ MB 1 0.2+ MB 1 0.121478 0.649902 1.623129 0.449454 0.380537 	Tools H 75341 no 75341 no 7541 no 75541 no 755541 no 755541 no 755541 no 755541 no 755541 no 755541 no 7555	in-null in-null in-null in-null in-null ibject(4) service	state FIN FIN FIN FIN FIN FIN FIN	saved spkts 6 14 8 10 10 	dpkts 4 38 16 6 6	sbytes 258 734 364 534 534	dbytes 172 42014 13186 268 268 	rate 74.087490 78.473372 14.170161 33.373826 39.417980 	••••	<pre>ct_dst_sport_ltm f f f f f f f f f f f f f f f f f f f</pre>	Ct_dst_src_ltm (1) Ct_dst_src_ltm (1) (1) (1) (1) (1) (1) (1) (1)	is_ftp_lo	Share Editing Gin ct_ft 0 0 0 0 0
File Edit V + Code + Te 40 ct 41 ct 42 i: 43 at 44 la dtypes: memory [5] data[da 0 1 2 4 5 175125	View Inseri Text tc_src_ltm tc_srv_dst is_sm_ips_; sttack_cat label s: float64; / usage: 6; data['serv: id 1 2 3 5 6 5 175126	t Runtime	Tools H 75341 no 75341 no 7534	m-null m-null m-null m-null m-null bject(4) service	state FIN FIN FIN FIN FIN FIN FIN FIN FIN	saved spkts 6 14 8 10 10 10	dpkts 4 338 116 6 6 8	sbytes 258 734 364 534 534 	dbytes 172 42014 13186 268 268 268 	rate 74.087490 78.473372 14.170161 33.373826 39.417980 39.417984 26.018748		<pre>ct_dst_sport_ltm f f f f f f f f f f f f f f f f f f f</pre>	RAM □ Disk □ ct_dst_src_ltm 1 1 2 3 40 40 … 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	is_ftp_lo	Share Control Contro Control Control Control Control Control Control Contro

Step3: Data cleansing, normalization, and null value detection

In Python Label Encoding, the category value must be changed to a numeric value in the range of 0 to the class labels' total number, minus one. if the outcome of the category variable.

Data Normalization

• Normalization of Data 58 Data Frame's numeric columns are scaled from 0 to 1 using the Min Max Scaler.

Binary Classification Preparation

- The Data Frame is duplicated for the Binary Classification process.
- The 'label' property is divided into two groups: 'normal' and 'abnormal'.

- Using Label Encoder (), the column 'label' is encoded, and the resulting encoded labels (0,1) are preserved in the column itself.
- 61 columns and 81173 rows in a binary dataset

Multi-class Classification Preparation

- The Data Frame is duplicated for the Multi-class Classification process.
- Nine categories are listed under the 'attack_cat' attribute: 'Analysis', 'Backdoor', 'DoS', 'Exploits', 'Fuzzers', 'Generic', 'Normal', 'Reconnaissance', and 'Worms'.
- Label Encoder () is used to encrypt attack_cat, and the accompanying encoded labels (0,1,2,3,4,5,6,7,8) are kept in the label attribute.
- attack_cat is encoded using one-hot.
- Multi-class Dataset 81173 rows, 69 columns

O File	Edit View Inst	ETECTION SYSTEM.ipy sert Runtime Tools Help	b ☆ All changes saved	Ę	Comment よ Share 🌣
+ Co	de + Text			RAM I	✓ ✓ Editing ∧
_				Disk =	
[7] ₀s	data.isnull()).sum()			
	id	0			
	dur	0			
	proto	0			
	service	94168			
	state	0			
	spkts	0			
	dpkts	0			
	sbytes	0			
	dbytes	0			
	rate	0			
	sttl	0			
	dttl	0			
	sload	0			
	dload	0			
	sloss	0			
	dloss	0			
	sinpkt	0			
	dinpkt	0			
	sjit	0			
	djit	0			
	swin	0			

Step4: Classify the type of attacks



Code + Text	Eile F	Edit Viev	v Insert	Runtime	Tools I	Help All c	hanges	saved							Commen	nt 👫 Share	4
9) data.dropna(inplace=True) 0 data.shape (81173, 45) 11) data['attack_cet'].value_counts() Generic 39496 Normal 19488 Exploits 16187 DoS 1791 Fuzers 1731 Reconnaissance 1793 Analysis 564 Norms 114 Backdoor 99 Name: attack_cat, dtype: int64 * Os completed at 10:21 AM Intrusion DETECTION SYSTEM.jpynb ☆ File Edit View Insert Runtime Tools Help All changes saved Code: + Text * Editing	Code	+ Text													✓ RAM I	Editing	
data.shape [5: (81173, 45) [11] data['attack_cat'].value_counts() Generic 39496 Normal 19488 Exploits 16187 DoS 1791 Fuzzers 1731 Reconnaissance 1783 Analysis 564 Worms 114 Backdoor 99 Name: attack_cat, dtype: int64	[9] d	data.drop	pna(inpl	ace=True)													
<pre>C (8173, 45) 11] data['attack_cat'].value_counts() G Genric 39496 Normal 19488 Exploits 16187 Do5 1791 Fuzzers 1731 Reconnaissance 1783 Analysis 564 Norms 114 Backdoor 99 Name: attack_cat, dtype: int64 12] data['state'].value_counts() C Os completed at 10:21 AM C INTRUSION DETECTION SYSTEM.ipynb ☆ File Edit View Insert Runtime Tools Help All changes saved Code + Text</pre>	0 d	data.shap	pe														
11] data['attack_cat'].value_counts() Generic 99496 Normal 19488 Exploits 16187 DoS 1791 Fuzzers 1731 Reconnaissance 1783 Analysis 564 Worms 114 Backdoor 99 Name: attack_cat, dtype: int64 12] data['state'].value_counts() I2] data['state'].value_counts() I2] file Edit View Insert Runtime Tools Help All changes saved Code + Text Image: dtstey, wyper another	C• (81173, 4	45)														
Generic 39496 Normal 19488 Exploits 16187 DoS 1791 Fuzzers 1731 Reconnaissance 1703 Analysis 564 Worms 114 Backdoor 99 Name: attack_cat, dtype: int64 12] data['state'].value_counts() ✓ Os completed at 10:21 AM ✓ INTRUSION DETECTION SYSTEM.ipynb ☆ File Edit View Insert Runtime Tools Help All changes saved Code + Text ✓ Pisk III ✓ Editing	11] d	data['ati	tack cat	'l.value d	ounts())											
	G N E D F R A W B N	Generic Normal Exploits DoS Euzzers Reconnais Nalysis Norms Backdoor Name: att	ssance tack_cat	39496 19488 16187 1791 1731 1703 564 114 99 , dtype: i	nt64												
	2] d File I Code	data['sta NTRUSIC Edit Vieu + Text data	on DETI N DETI w Insert	ECTION S Runtime	() SYSTEM Tools	1.ipynb Help <u>All c</u>	☆ .hanges	✓ saved	Os co	ompleted	at 10:21 A	M			Commer	nt 🔹 Share	•
C ⁺ id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_	d d L2] d L2 L2 d L2 L2 L2 L2 L2 L2 L2 L2 L2 L2	data['sta NTRUSIC Edit Vien + Text data	ate'].va DN DET w Insert	Lue_counts ECTION S Runtime	s() SYSTEM Tools	1.ipynb Help <u>All c</u> service	☆ <u>state</u>	✓ saved spkts	Os co dpkts	sbytes	at 10:21 A dbytes	M		ct_dst_sport_ltm	RAM I Commer	nt ـــــــــ Share	e ct,
id dur proto state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_dst_sport_ltm is_ftp_login is_ftp_login <th< th=""><td>2] d L File D Code</td><td>data['sta NTRUSIC Edit Vien + Text data 3</td><td>ate'].va DN DETI w Insert tate, uty id 4</td><td>Lue_counts ECTION S Runtime per ancor dur 1.681642</td><td>s() SYSTEM Tools proto tcp</td><td>1.ipynb Help <u>All c</u> service ftp</td><td>☆ hanges state FIN</td><td>saved spkts 12</td><td>Os co dpkts 12</td><td>sbytes 628</td><td>at 10:21 A dbytes 770</td><td>NM rate 13.677108</td><td>••••</td><td>ct_dst_sport_ltm 1</td><td>RAM Commer Disk m •</td><td>nt 🕰 Share Fediting is_ftp_login 1</td><td>e ct</td></th<>	2] d L File D Code	data['sta NTRUSIC Edit Vien + Text data 3	ate'].va DN DETI w Insert tate, uty id 4	Lue_counts ECTION S Runtime per ancor dur 1.681642	s() SYSTEM Tools proto tcp	1.ipynb Help <u>All c</u> service ftp	☆ hanges state FIN	saved spkts 12	Os co dpkts 12	sbytes 628	at 10:21 A dbytes 770	NM rate 13.677108	••••	ct_dst_sport_ltm 1	RAM Commer Disk m •	nt 🕰 Share Fediting is_ftp_login 1	e ct
id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_dst_src_ltm is_ftp_login ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login is_ftp_login ct_dst_src_ltm is_ftp_login ct_dst_src_ltm is_ftp_login is_ftp_login is_ftp_login is_f	2] d ▲ IN File I Code	data['sta NTRUSK Edit View + Text data 3 11	ate'].va DN DETI w Insert id 4 12	Lue_counts ECTION S Runtime pe: 20007 1.681642 2.093085	s() SYSTEM Tools proto tcp tcp	1.ipynb Help All d service ftp smtp	☆ thanges state FIN FIN	spkts 12 62	0s co dpkts 12 28	sbytes 628 56329	at 10:21 A dbytes 770 2212	M rate 13.677108 42.520967		ct_dst_sport_ltm 1 1	RAM Commer RAM Commer Ct_dst_src_ltm d 3 2	nt 🛣 Share Fediting is_ftp_login 1 0	e ct
id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_ltm ct_dst_src_ltm is_ftp_login ct_dst_src_ltm 3 4 1.681642 tcp ftp FIN 12 628 770 13.677108 1 3 1 11 12 2.093085 tcp smtp FIN 62 28 56329 2212 42.520967 1 2 0 0 15 16 0.000002 udp snmp INT 2 0 138 0 500000.001300 1 4 0	2] d ▲ IN File I Code	Adata['sta NTRUSIC Edit View + Text data 3 11 15	id 12 16	Lue_counts ECTION S Runtime J.c. 2003 dur 1.681642 2.093085 0.000002	s() SYSTEM Tools proto tcp tcp tcp udp	1.ipynb Help <u>All c</u> service ftp smtp snmp	☆ state FIN FIN INT	spkts 12 62 2	0s co dpkts 12 28 0	sbytes 628 56329 138	dbytes 770 2212 0	M rate 13.677108 42.520967 50000.001300		ct_dst_sport_ltm 1 1 1	RAM Disk Commer ct_dst_src_ltm S 3 2 4	nt 🕹 Share Editing is_ftp_login 1 0 0	2
id dur proto service state spkts dpkts sbytes dbytes rate ct_dst_sport_1tr ct_dst_src_1tr is_ftp_login ct_dst_src_1tr 3 4 1.681642 tcp ftp FIN 12 628 770 13.677108 1 3 1 11 12 2.093085 tcp smmp FIN 62 28 56329 2212 42.520967 1 2 0 1 15 16 0.00002 udp snmp INT 2 0 138 0 50000.001300 1 4 0 17 18 0.393556 tcp http FIN 10 8 860 1096 43.195866 1 2 0	d d Code D	data['sta NTRUSIC Edit View + Text data 3 11 15 17	ate'].va	Lue_counts ECTION S Runtime pe: ancor dur 1.681642 2.093085 0.000002 0.393556	s() SYSTEM Tools proto tcp tcp udp tcp	1.ipynb Help <u>All c</u> service ftp smtp snmp http	state FIN FIN INT FIN	saved spkts 12 62 2 10	0s co dpkts 12 28 0 8	sbytes 628 56329 138 860	dbytes 770 2212 0 1096	M rate 13.677108 42.520967 500000.001300 43.195886		ct_dst_sport_ltm 1 1 1 1	RAM Commer RAM Disk Commer ct_dst_src_ltm S 3 2 4 2	nt 🛣 Share Fditing is_ftp_login 1 0 0 0	e ct
id dur proto start spkts dpkts sbytes dbytes rate ct_dst_sport_int ct_dst_src_int is_ftp_login ct_dst_sport_int 3 4 1.681642 tcp ftp FlN 12 12 628 770 13.677108 1 3 1 3 1 11 12 2.093085 tcp smtp FlN 62 28 56329 2212 42.52067 1 2 0 0 15 16 0.00002 udp smmp INT 2 0 138 0 50000.001000 1 4 0 17 18 0.393556 tcp http FlN 10 8 860 1096 43.195886 1 1 0 12 2.0.38017 tcp http FlN 10 6 98 268 44.376488 1	d ▲ IN File I Code Code	data['sta NTRUSIC Edit View + Text data 3 11 15 17 21	id 12 16 18 18 12	Lue_counts ECTION S Runtime dur 1.681642 2.093085 0.000002 0.393556 0.338017	s() SYSTEN Tools proto tcp tcp tcp tcp tcp	1.ipynb Help All c service ftp smtp snmp http	☆ state FIN INT FIN FIN FIN	✓ saved spkts 12 62 2 10 10 10	0s co dpkts 12 28 0 8 8 6	sbytes 628 56329 138 860 998	dbytes 770 2212 0 1096 268	M rate 13.677108 42.520967 500000.001300 43.195886 44.376468		ct_dst_sport_ltm 1 1 1 1 1 1 1 1	RAM Commer Disk Commer ct_dst_src_ltm S 3 2 4 4 2 1	nt X Share Fediting is_ftp_login 1 0 0 0 0 0	ct_
id du proto service state spkts dpkts sbytes dpkts rate ct_dst_sport_int ct_dst_src_int is_ftp_login ct_dst_src_int 3 4 1.681642 tcp ftp ftp 12 628 770 13.677108 1 3 1 11 12 2.093085 tcp smp FIN 62 28 56329 2212 42.520967 1 2 0 0 15 16 0.00002 udp smmp INT 2 0 138 0 50000.001000 1 4 0 17 18 0.393556 tcp http FIN 10 8 860 1096 43.195866 1 1 0 10 12 0.388077 tcp http FIN 10 6 98 268 44.376468 1 1	d ▲ IN File I Code Code	data['sta NTRUSIC Edit View + Text data 3 11 15 17 21 	id 12 16 18 18 18 18 18 18	Lue_counts ECTION S Runtime dur 1.681642 2.093085 0.00002 0.393556 0.338017 	s() SYSTEN Tools proto tcp tcp tcp tcp tcp	1.ipynb Help <u>All c</u> service ftp smtp http http	☆ state FIN FIN FIN FIN FIN	spkts spkts 12 62 2 10 10	0s cc dpkts 12 28 0 8 6 	sbytes 628 56329 138 860 998 	dbytes 770 2212 0 1096 268 	M rate 13.677108 42.520967 50000.001300 43.195886 44.376468		ct_dst_sport_ltm 1 1 1 1 1 1 	Commer	nt The Share	e i
id du proto servic state spkto gpkto spkto gpkto spkto spkt	d A IN File I Code	data['sta NTRUSIC Edit Vien + Text data 3 11 15 17 21 175335	id 175336	Lue_counts Runtime per 2000 dur 1.681642 2.093085 0.000002 0.393556 0.338017 0.000006	s() SYSTEN Tools Proto tcp tcp tcp tcp tcp tcp tcp	1.ipynb Help All c service ftp smtp snmp http http dns	☆ state FIN FIN FIN FIN FIN FIN	spkts spkts 12 62 10 10 10 2 2 2 2 2 2 3 2 3 4 3 4	0s cr dpkts 12 28 0 8 6 0	sbytes 628 56329 138 860 998 114	dbytes 770 2212 0 1096 268 0	M rate 13.677108 42.520967 50000.001300 43.195886 44.376468 166666.60800		ct_dst_sport_ltm 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Comment RAM → Disk → ct_dst_src_ltm 3 2 4 2 4 2 1 45	tis_ftp_login 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	e ct_
id du proto service state spkte dpkte sbytes dpkte rate r.r. ct_dst_sport_ite ct_dst_src_1te is_ftp_login ct_dst_sport_ite 3 4 1.681642 1cp ftp	d d File I Code €	Adata ['sta NTRUSIC Edit View + Text data 3 11 15 17 21 175335 175336	id 175336 175337	Lue_counts Runtime pe: 10005 dur 1.681642 2.093085 0.0393556 0.338017 0.000006 0.000009	s() SYSTEN Tools tcp tcp tcp tcp tcp tcp tcp tcp tcp tcp	1.ipynb Help All c service ftp snmp http http http dns dns	☆ state FIN FIN FIN FIN FIN TINT	spkts 12 62 10 10 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	0s cr dpkts 12 28 0 8 6 0 0 0 0 0	sbytes 628 56329 138 860 998 114 114	dbytes 770 2212 0 1096 268 0 0	M rate 13.677108 42.520967 50000.001300 43.195886 44.376468 1666666.660800 111111.107200		ct_dst_sport_ltm 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Comment RAM → Disk → ct_dst_src_ltm 3 3 2 4 2 1 → 45 24	ts_ftp_login is_ftp_login 0 0 0 0 0 0 0 0 0 0 0 0 0	ct_
id du prote sterve	12] d File Code	data['sta NTRUSIC Edit View + Text data 3 11 15 17 21 175335 175336 175338	ate'].va	Lue_counts Runtime CTION S Runtime L.681642 2.093085 0.000002 0.393556 0.338017 0.000006 0.000009 0.000009	s() SYSTEN Tools proto tcp tcp tcp tcp tcp tcp tcp tcp tcp udp udp udp	1.ipynb Help <u>All c</u> service ftp smtp http http http dns dns dns	☆ state FIN FIN FIN FIN FIN FIN FIN TIN TINT	spkts 12 62 10 10 2	0s cr dpkts 12 28 0 8 6 0 0 0 0 0 0 0	sbytes 628 56329 138 860 998 114 114 114	at 10:21 Å dbytes 770 2212 0 1096 268 0 0 0 0	M rate 13.677108 42.520967 50000.001300 43.195886 44.376468 166666.660800 111111.107200 111111.107200		<pre>ct_dst_sport_ltm</pre>	Commer RAM ✓ Disk ✓ ct_dst_src_ltm 3 2 4 2 1 45 24 13	nt X Share Editing is_ftp_login 1 0 0 0 0 0 0 0 0 0 0 0 0 0	ct_
id du proto service state spkte dpkte spkte dpkte rate ct_dst_sport_ite ct_dst_src_ite is_ftp_login ct_dst_sport_ite 3 4 1.681642 tcp ftp	12] d File Code E	data['sta NTRUSIC Edit View + Text data 3 11 15 17 21 175335 175336 175338 175338	ate'].va	Lue_counts ECTION S Runtime dur 1.681642 2.093085 0.000002 0.393556 0.338017 0.000006 0.000009 0.000009 0.000009	s() SYSTEN Tools proto tcp tcp tcp tcp tcp tcp tcp tcp tcp udp udp udp udp udp	1.ipynb Help All c service ftp smtp http http http dns dns dns dns dns	☆ state FIN FIN FIN FIN FIN FIN INT INT	saved spkts 12 62 2 10 10 10 2	dpkts 12 28 0 8 6 0 0 0 0 0 0 0 0	sbytes 628 56329 138 860 998 114 114 114 114	dbytes 770 2212 0 1096 268 0 0 0 0 0	M rate 13.677108 42.520967 500000.001300 43.195886 44.376468 166666.660800 111111.107200 111111.107200		ct_dst_sport_ltm 1 1 1 1 1 1 1 1 1 1 1 1 1		nt * Share Editing is_ftp_login 1 0 0 0 0 0 0 0 0 0 0 0 0 0	e 1

Step5: Feature Extraction:

This is the procedure of reducing the number of variables in a statistical model. The number of variables should be kept to a minimum to decrease overall computation costs and, in some cases, to improve model performance. The input parameters with the strongest correlation to the target attribute are selected after statistical analysis is used to assess the relationship between each input parameter and the goal parameter. Although the statistical measures utilized are influenced by the type of data in both the dependent and independent variables, these processes can be simple and efficient. The random forest gets estimates from each decision tree and calculates the right answer based on the number of votes cast, as opposed to relying just on one decision tree. Compared to other techniques, training takes less time with this one. Even with a large dataset, it runs quickly and

accurately predicts the outcome. Even in the absence of a sizable volume of data, accuracy can still be maintained.

Selection of features

- 'bin_data' has 61 characteristics total.
- 'multi_data' has 69 properties total.
- For feature extraction, the Pearson Correlation Coefficient approach is employed.
- •
- The attributes with a correlation coefficient of more than 0.3 to the target attribute label were chosen.
- The attributes with a correlation coefficient of more than 0.3 to the target attribute label were chosen
- Number of 'bin_data' attributes after feature selection: 15
- "rate,""sttl,""sload,""dload,""ct_srv_src,""ct_state_ttl,""ct_dst_ltm,""ct_dst_src_ltm,""ct_src_ ltm,""ct_srv_dst," "state_CON," "state_INT," and "label."
- After feature selection, the number of "multi_data" attributes is 16
- "dttl," "swin," "dwin," tcprtt," synack," ackdat," label," proto_tcp,"proto_udp,"service_dns," state_CON,"state_FIN,"attack_cat_Analysis,"attack_cat_DoS,"attack_cat_Exploits,"andattac k_cat_Normal."

2) 🝐 File	INTRUSION DETECTION SYSTEM.ipynb 😒 Edit View Insert Runtime Tools Help <u>All changes saved</u>	Comment	👪 Share	¢
:	+ Coo	de + Text	V RAM I	Editing	^
}	[17]	<pre># selecting column names of all data types nominal_names = features['Name'][features['Type ']=='nominal'] integer_names = features['Name'][features['Type ']=='integer'] binary_names = features['Name'][features['Type ']=='binary'] float_names = features['Name'][features['Type ']=='float']</pre>			
1	[18]	<pre># selecting common column names from dataset and feature dataset cols = data.columns nominal_names = cols.intersection(nominal_names) integer_names = cols.intersection(integer_names) binary_names = cols.intersection(binary_names) float_names = cols.intersection(float_names)</pre>			
*	[19]	<pre># Converting integer columns to numeric for c in integer_names: pd.to_numeric(data[c])</pre>			
9 2 9	[20]	<pre># Converting binary columns to numeric for c in binary_names: pd.to_numeric(data[c])</pre>			
		· · · · · · · · · · · · · · · · · · ·			•

Step 6: Splitting the Dataset into the Train & Test sets:

we separate our sample into a test set and a train set. This important data pre-processing step increases our classification model's efficacy. Suppose we train our classification model on a single sample and validate it on a second sample. Consequently, our model will struggle to comprehend

how the two models relate to one another. The model's performance will deteriorate if we give it a new dataset after it has received enough training and has a high training accuracy. A substantial amount of effort has been devoted to the development of a classification model that exhibits strong performance on the test & training datasets alike. The train set was included in the sample used to develop the classification model, and the result is now known for these samples. The term "test set" refers to a subset of the sample that is used to forecast output when the model is tested.

Splitting Dataset into Training and Testing

- Randomly splitting the **bin_data** in **80% for training** and **20% for testing**.
- Randomly splitting the **multi_data** in **70% for training** and **30% for testing**.
- Target feature *label*

								+ 0	Code - +	Text						
30] dat	ta_c	at.head()													
	pr	roto_tcp	proto_udp	servi	ce_dhcp	service_	dns service	ftp ^s	ervice_ftp dat	service_h	http	service_irc	service_pop3	service_radiu	s service_sm	ntp
3	3	1	0		0		0	1		1	0	0	0	1)	
11	1	1	C		0		0	0			0	0	0)	
15	5	0	1		0		0	0	-)	0	0	0)	
17	7	1	0		0		0	0		1	1	0	0)	
21	1	1	0		0		0	0)	1	0	0	1)	
lINT File Edi Code	TRUS dit V + Te	SION DE /iew Inse ext	TECTION ert Runtime	SYSTE Tools	M.ipynb Help <u>A</u>) ☆ .ll changes	<u>saved</u>						~	Commen	t 🔉 Share	
Code	TRUS dit V + Te	SION DE /iew Inse ext	TECTION ert Runtime	SYSTE Tools	M.ipynb Help <u>A</u>) 1 Il changes	<u>saved</u>						~	RAM Disk	t 🕹 Share	
INT File Edi Code	TRUS dit V + To usin inmax ef no for ar df retu	g minmax c_scale = rmalizat i in col r = df[i rr = np.a [i] = mi rn df	scaler for MinMaxSca ion(df,col] mray(arr) nmax_scale	SYSTE Tools r norma ler(fea): .fit_tr	M.ipynb Help <u>A</u> Dizing c Dive_rar) 📩 data nge=(0, 1 farr.resh	<pre>saved)) ape(len(arr),</pre>	1))					~	Commen Disk	t 🕹 Share	9
INT File Edi Code # u min det 1 38] # c dat	IRUS dit V + To usin inmax af no for ar df retu data data.h	g minmax _scale = rmalizat i in col rr = df[i rr = np.a [i] = mi rn df	rt Runtime scaler for MinMaxSca ion(df,col : Jurray(arr) nmax_scale	SYSTE Tools r norma ler(fea): .fit_tr ion	M.ipynb Help A alizing o ture_rar) ☆ lata lge=(0, 1	<pre>saved)) ape(len(arr),</pre>	1))					~	Commen RAM Disk - •	Editing	e
INT File Edi Code # c min def t r r s 388] # c dat	TRUS dit V + Tu usin inmax af no for ar df retu data ata.h	sion DE fiew Inse ext g minmax cscale = rmalizat i in col rr = df[i rr = df[i rr = df[i rr = df[i r = df[i] = mi r = df[i r = df[i] = mi r = df[i] = mi	rtection ert Runtime scaler for MinMaxSca ion(df,col] rray(arr) nmax_scale normalizat	SYSTE Tools r norma ler(fea): .fit_tr ion dpkts	M.ipynb Help <u>A</u> blizing c sture_ran) ☆ lata lge=(0, 1 'arr.resh	<pre>saved)) ape(len(arr), rate</pre>	1)) sttl	dtt1	sload		service_radi	✓ us service_su	Commen RAM Disk · ·	t ≚ Share ✓ Editing	8
INT File Edi Code	TRUS usin inmax ef no for ar af retu data ata.h in 3	g minmax c_scale = rmalizat i in col rr = df[i rr = np.a [[i] = mi h before lead() d d 4 1.6816	scaler for MinMaxScaler ion(df,col] mray(arr) normalizat Mur spkts 42 12	SYSTE Tools r norma ler(fea); .fit_tr ion dpkts 12	M.ipynb Help A alizing o ature_rar cansform(sbytes 628) ☆ Iata age=(0, 1 'arr.resh dbytes 770	<u>saved</u>)) ape(len(arr), rate 13.677108	1))) sttl 62	dtt1 252 2.7	sload 10179e+03		service_radi	us service_su 0	Commen RAM Constant Disk Commen	a Share ✓ Editing ap service_s 0	8
INT File Edi Code	TRUS ddit V + To usin inmax af no for ar ar df retu data ata.h in 3 4 11 12	sion DE fiew Inse ext g minmax cscale = rmalizat i in col rr = df[i rr = df[i rr = df[i r = df[i r = df[i d d 4 1.6816 2 2.0930	rtection ert Runtime scaler for MinMaxSca ion(df,col :] rray(arr) nmax_scale normalizat ur spkts 42 12 85 62	SYSTE Tools r norma ler(fea): .fit_tr ion dpkts 12 28	M.ipynb Help A hlizing c hture_ran ransform(sbytes 628 56329) ☆ lata lage=(0, 1 'arr.resh dbytes 770 2212	<pre>saved)) ape(len(arr), rate 13.677108 42.520967</pre>	1))) sttl 62 62	dtt1 252 2.7 252 2.1	sload 10179e+03 18251e+05	····	service_radi	us service_su 0 0	Commen RAM Disk	Editing P Editing	5



Step7: Comparison of Linear Regression, Logistic Regression and RL.

Linear regression models

It is possible to look at the link between one or more independent variables and a continuous dependent variable. If there is only one independent variable and one dependent variable, then simple linear regression is used. When there are more than two factors, though, multiple linear regression is used instead of simple linear regression. In all types of linear regression, the least squares method is used to find the line that fits a set of data the best. Logistic regression, like linear regression, can be used to foretell numbers for categorical variables. This method is used to evaluate the correlation between several predictors and a single outcome. Categorical variables can take on the values true, false, yes, no, 1, 0, and other values. The S-curve is transformed into a straight line by the logit

function, and the unit of measurement in linear regression is a probability. In regression analysis, both models are used to make predictions about the future. However, linear regression is often easier to understand. In addition, whereas a smaller sample size is sufficient for linear regression to accurately represent values across all answer categories, a larger sample size is necessary for logistic regression. Should a more extensive, representative sample be lacking, the model's statistical potential to identify a noteworthy effect may be compromised.

Linear Regression Model

Binary Classification

- Accuracy 97.80720665229443
- Mean Absolute Error **0.021927933477055742**
- Mean Squared Error 0.021927933477055742
- Root Mean Squared Error 0.1480808342664767
- R2 Score **88.20923868071647**

Multi-class Classification

- Accuracy 95.12976346911958
- Mean Absolute Error 0.06824901445466491
- Mean Squared Error 0.12146846254927726
- Root Mean Squared Error 0.3485232596962178
- R2 Score 91.82055676180129

LINEAR REGRESSION [67] [67] [7] [68] lr_bin = LinearRegression(normalize=False) [1r_bin.fit(X_train, y_train) [4] [68] /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_base.py:155: FutureWarning: 'normalize' was deprecated in version 1.0 and will be removed FutureWarning, LinearRegression(normalize=False) [4] [69] y_pred = lr_bin.predict(X_test)



Logistic regression

Among the family of supervised machine learning models is logistic regression. It is also considered a discriminative model, meaning that it aims to differentiate between multiple classes. Naive Bayes and other generative algorithms can make examples of the class they are trying to predict, but this one can't (for example, a cat picture). We already talked about how the model's beta values are made by logistic regression, which maximises the log likelihood function. This slightly alters when looking at it from a machine learning standpoint. In machine learning, the global maximum is found using gradient descent, with the loss function being the negative log likelihood. This is merely a different method to get the same approximations as before. Moreover, logistic regression may lead to overfitting, especially if the model has a large number of predictor variables. In high dimensionality models, regularisation is usually employed to penalise parameters with large coefficients. For additional information on the logistic regression machine learning model, go visit Scikit-learn (link is external to IBM).

Logistic Regression Model

Binary Classification

- Accuracy 97.80104712041884
- Mean Absolute Error **0.02198952879581152**
- Mean Squared Error 0.02198952879581152
- Root Mean Squared Error 0.1482886671186019
- R2 Score 88.17947258428785

Multi-class Classification

- Accuracy 97.58952036793693
- Mean Absolute Error **0.060077201051248356**
- Mean Squared Error 0.18056011826544022
- Root Mean Squared Error 0.42492366169165047
- R2 Score 87.87674567880146

>	▲ File	INTRUSION DETECTION SYSTEM.ipynb 🏠 Edit View Insert Runtime Tools Help <u>All changes saved</u>	Comment	👪 Share	¢
-	⊦ Cod	le + Text	V RAM Disk	🖍 Editing	^
	LOG	ISTIC REGRESSION			
V os	[78]	logr_bin = LogisticRegression(random_state=123, max_iter=5000) logr_bin			
		LogisticRegression(max_iter=5000, random_state=123)			
1 5	[79]	<pre>logr_bin.fit(X_train,y_train)</pre>			
		LogisticRegression(max_iter=5000, random_state=123)			
V Os	[80]	<pre>y_pred = logr_bin.predict(X_test)</pre>			
V os	[81]	<pre>print("Mean Absolute Error - " , metrics.mean_absolute_error(y_test, y_pred)) print("Nean Squared Error - " , metrics.mean_squared_error(y_test, y_pred)) print("Root Mean Squared Error - " , np.sqrt(metrics.mean_squared_error(y_test, y_pred))) print("R2 Score - " , metrics.explained_variance_score(y_test, y_pred)*100) print("Accuracy - ",accuracy_score(y_test,y_pred)*100)</pre>			



Logistic regression is commonly applied to problems involving classification and forecasting through the use of cases. These use cases consist of:

•Fraud detection:

Teams can use logistic regression models to find data abnormalities that point to fraudulent activity. Banking and other financial organisations may discover that specific behaviours or qualities are more often linked to fraudulent operations in an effort to better protect their consumers. SaaS-based organisations also employ these techniques to eliminate fictitious user accounts from datasets during business performance data analysis.

•Disease prediction:

This analytics technique can be implemented in medicine to predict the likelihood that a given population will become unwell or sick. Health care facilities can arrange for prophylactic care for individuals who are more susceptible to a specific disease.

•Churn prediction:

Certain behaviours may signal churn in different organisational activities. If high achievers are in danger of leaving the organisation, human resources and management teams may be interested in learning about it. Information of this kind could start conversations regarding the company's compensation policies or culture. Alternatively, the sales staff would like to know which of its clients might choose to work with another company. Teams might be inspired by this to develop a retention plan in order to prevent revenue loss.

Types of logistic regression Logistic regression models based on categorical responses come in three varieties.

•Binary logistic regression:

This method's dependent variable, or answer, is dichotomous, meaning that there are only two possible outcomes (e.g., 0 or 1). It can determine the malignancy of a tumour or whether an email is spam, among other applications. This technique is widely used in the field of logistic regression and is widely recognised as a top-tier classifier for binary data.

•Multinomial logistic regression:

Within this particular variant of logistic regression, the dependent variable is open to a maximum of three values, none of which are sequentially arranged. To improve their ability to sell their films, for instance, film studios endeavour to predict the genre of film that a given audience is most likely to enjoy. To determine the degree to which age, relationship status, gender, and age influence the genre of film that an individual prefers, the studio may implement a multinomial logistic regression model. As a result, the studio may opt to focus its marketing endeavours on the specific demographic that is relevant to a given film.

•Ordinal logistic regression:

Despite the set order of values shown, this type of logistic regression model is used when the response variable has three or more possible outcomes. Ordinal responses can be expressed as grading schemes or rating scales ranging from 1 to 5.



Multi-class Classification

- Accuracy 97.59362680683311
- Mean Absolute Error **0.059912943495400786**
- Mean Squared Error 0.17941031537450722
- Root Mean Squared Error 0.42356854861345317
- R2 Score 87.93449282205455

• SVM algorithm

- **1. Input:** The algorithm takes a labelled dataset as input, where each data point consists of a set of features (X) & a corresponding binary class label (Y) indicating one of two classes (e.g., 0 or 1).
- **2. Data Preparation:** If needed, feature scaling and data normalization can be performed to ensure that all features have a similar scale, which can improve the convergence and performance of the algorithm.
- **3. Feature Space Transformation Kernel Trick Optional:** The initial feature space can be changed into a higher-dimensional space using a kernel function in SVM. This enables SVM to identify a decision boundary that may have been nonlinear in the original space but is now linear in the modified space. The linear kernel, polynomial kernel, sigmoid kernel, and radial basis function RBF kernel are examples of common kernel functions.
- 4. Margin Maximization: Finding the hyperplane with the largest margin between the two classes is the primary goal of SVM. The margin is the distance between the hyperplane and the closest data points of each class support vectors. The optimal hyperplane is the one that achieves the largest margin while correctly classifying the data points.
- **5. Soft Margin Optional**: SVM introduces slack variables to allow for a soft margin in situations where the data is not perfectly separable. These slack variables allow some data points to be misclassified or fall within the margin region, but with a penalty for misclassification. The objective is to balance the margin maximization and the classification error.
- 6. Optimization Problem: The SVM technique frames the issue as a convex optimisation problem in order to determine the ideal weights and biases for the hyperplane. The optimisation seeks to maximise the margin and minimise the classification error of incorrectly categorised data points.
- **7. Kernel Parameters Optional:** In the event that a kernel function is employed, the SVM algorithm may further entail fine-tuning the kernel parameters, such as the polynomial kernel's degree or the RBF kernel's width, in order to maximise performance on the training set.
- **8. Training:** In the training phase, the algorithm finds the ideal hyperplane parameters by solving the optimisation issue using methods such as quadratic programming or Sequential Minimal Optimisation (SMO).
- **9. Prediction:** The model can be used to predict the class label for fresh data points once it has been trained. The predicted class label is determined by comparing the signed distance of the data point to the hyperplane. If the signed distance is positive, the data point belongs to one class; otherwise, it belongs to the other class.
- **10. Output:** For binary classification tasks, the Support Vector Machine technique provides an output of the learnt model parameters (weights and bias). It is well known that SVM can handle high-dimensional data and find difficult decision boundaries, making it a reliable classification

system. When the data are clear and the categories are distinct, this method shines. Using techniques such as One-vs-One (OvO) or One-vs-Rest OvR strategies, SVM can be adapted to perform multiclass classification. In addition, the Support Vector Regression SVR variant of SVM can be used for regression assignments.

4. RESULTS AND DISCUSSION

The assessment of the adaptive neural network approach to intrusion detection involved conducting three experiments using the prototype application. The effectiveness of several algorithms, such as SVM, linear regression, & logistic regression, was assessed in these trials.

The results clearly demonstrated that the support vector machine consistently provided higher accuracy compared to both linear and logistic regression models. This suggests that the adaptive neural network approach, when implemented using a support vector machine, holds significant promise for improving IDS ' effectiveness and accuracy.

5. CONCLUSION

IDS are essential for protecting computer and network-based systems because they continuously monitor protocol traffic for possible breaches and assaults. The adaptability of IDS allows for adjustments in response to both external and internal threats. In this study, we conducted a comparative analysis of IDS using linear logistic regression & SVM.

In conclusion, the experiments conducted using the prototype application to assess the viability of the adaptive neural network approach for intrusion detection revealed valuable insights. Specifically, the support vector machine algorithm consistently outperformed linear and logistic regression models in terms of accuracy. This finding underscores the potential effectiveness of the adaptive neural network approach for enhancing IDS.

These findings imply that more investigation and advancement in this area may result in IDS that are more reliable and accurate. Implementing adaptive neural networks, particularly with SVM, holds promise for addressing the evolving challenges in cybersecurity and bolstering our ability to detect and respond to intrusions effectively. However, additional investigations and real-world testing are warranted to fully harness the capabilities of this approach and validate its practicality in various security contexts.

ACKNOWLEDGMENT

In America, the word "acknowledgment" is most often spelled without a "g" following the "e." Steer clear of the stuttering phrase "One of us (R. B. G.) thanks." Say "R.B.G. thanks..." instead. Include acknowledgements of sponsors in the first page's unnumbered footnote.

REFERENCES

- Aburomman, A. A., &Reaz, M. B. I. (2016) "Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection."Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC): 636-640.
- [2] Al-Jarrah, O. Y., Al-Hammdi, Y., Yoo, P. D., Muhaidat, S., & Al-Qutayri, M. (2018) "Semi-supervised multilayered clustering model for intrusion detection." Digital Communications and Networks 4(4): 277-286.

- [3] Al-Yaseen, W. L., Othman, Z. A., &Nazri, M. Z. A. (2017) "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system." Expert Systems with Applications 67(1): 296-303.
- [4] An, X., Su, J., Lü, X., & Lin, F. (2018) "Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system." EURASIP Journal on Wireless Communications and Networking 249 (1): 1-9.
- [5] Belavagi, M. C., &Muniyal, B. (2016) "Performance evaluation of supervised machine learning algorithms for intrusion detection." Procedia Computer Science 89(1): 117-123.
- [6] Elrawy, M. F., Awad, A. I., &Hamed, H. F. (2018) "Intrusion detection systems for IoT-based smart environments: a survey." Journal of Cloud Computing 7 (1): 21
- [7] Elsaeidy, A., Munasinghe, K. S., Sharma, D., &Jamalipour, A. (2019) "Intrusion detection in smart cities using Restricted Boltzmann Machines." Journal of Network and Computer Applications 135 (1): 76-83.
- [8] Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., &Kannan, A. (2013) "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey." EURASIP Journal on Wireless Communications and Networking (1): 271.
- [9] Gautam, R. K. S., &Doegar, E. A. (2018) "An Ensemble Approach for Intrusion Detection System Using Machine Learning Algorithms." International Conference on Cloud Computing, Data Science & Engineering (Confluence): 14-15.
- [10] Ghasempour, A. (2019). Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges. Inventions 4(1): 22.