تشخیص حملات: یک روش خصمانه یادگیری ماشین

طه اخلاق‌پسندی، محمدهادی علائیان*

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی m.alaeiyan@kntu.ac.ir

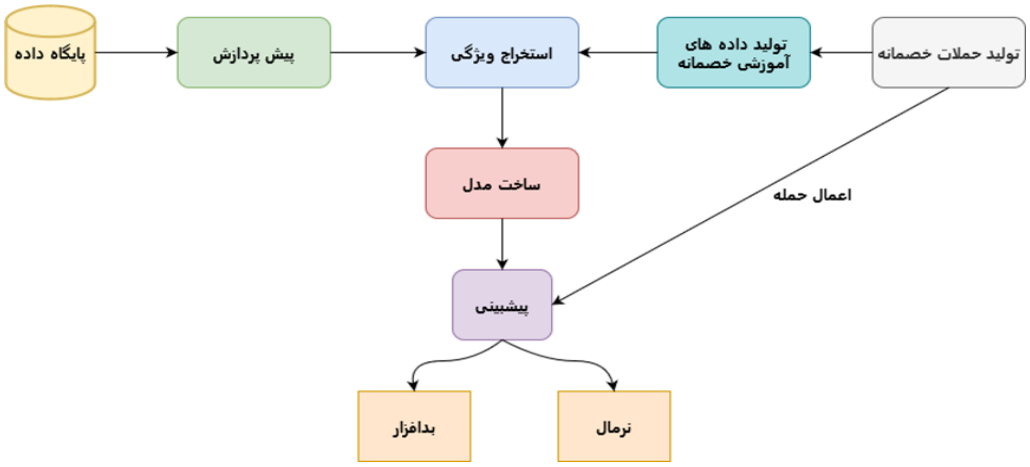
چکیده

امروزه روش‌های یادگیری ماشین در حوزه‌های گوناگون در حال استفاده است و تحلیل حملات از این موضوع مستثنی نیست. روزانه انواع مختلفی از حملات اعمال می‌گردد. به همین دلیل بررسی تک تک آن‌ها توسط نیروهای انسانی متخصص به دلیل کم بود تعداد متخصص، نسبت به تعداد رو به افزایش حملات و همچنین امکان وقوع خطای انسانی در تشخیص حملات، کاری خسته‌کننده و تقریبا ناممکن است. در سال‌های گذشته کارهای زیادی برای طراحی یک مدل یادگیری ماشین یا یادگیری عمیق برای تشخیص نفوذ انجام شده است. این مدل‌ها با استفاده از الگوریتم های یادگیری ماشین، مانند RF[1]، SVM[2]، Decision tree، Logistic Regression، Nave Bayes، DNN[3]، ANN[4]، CNN[5]، RNN[6]، LSTM[7] و GRU[8] با دقت های مختلفی ساخته شده است. گروهی با استفاده از یادگیری ماشین، یا یادگیری عمیق، مدل های مختلف با دقت های مختلف ساختند. در تمامی موارد، دقت به میزان خوبی بدست آمده است، اما هیچ کدام از آن‌ها مدل خودشان را در معرض حمله قرار نداده بودند. به عبارت بهتر، به مدلی که طراحی کرده بودند حمله نشده تا توانایی مدل خودشان را مورد ارزیابی قرار دهند.

هدف از این پژوهش، ارائه روشی برای بهبود نتیجه تشخیص نفوذ با استفاده از روش‌های یادگیری ماشین است. چرا که روش‌های یادگیری ماشین، در حال رشد هستند و دائماً با روش‌هایی که از لحاظ عملکردی و قدرت پردازشی، کارایی و دقت بهتری دارند جایگزین می‌شوند. در روش پیشنهادی ما علاوه بر ساخت یک مدل قابل قبول با داشتن دقت مطلوب، با استفاده از روش‌های حمله خصمانه، به مدل خود حمله می‌کنیم. شبکه های عصبی GAN[9] به عنوان یکی از چارچوب‌های قابل بهره‌وری برای اعمال حملات خصمانه، شامل مدل‌های مولدی هستند که داده‌های جدید شبیه داده‌های آموزشی تولید می‌کنند.

کلمات کلیدی: حملات ، یادگیری ماشین ، یادگیری عمیق

[1] Random Forest
[2] Support vector machine
[3] Deep Neural Network
[4] Artificial neural network
[5] Convolutional Neural Networks
[6] Recurrent Neural Networks
[7] Long short-term memory
[8] Gated Recurrent Units
[9] Generative Adversarial Network
* Corresponding author (m.alaeiyan@kntu.ac.ir)



شکل ۱- دیاگرام بلوکی از روند پیشنهادی برای انجام پروژه

مقدمه

مساله اصلی در این مقاله، ارائه راه کارهای موجود، برای بهبود نتیجه تشخیص نفوذ با استفاده از روش‌های یادگیری ماشین است. چرا که روش‌های یادگیری ماشین، در حال رشد هستند و دائماً با روش‌هایی که از لحاظ عملکردی و قدرت پردازشی، کارایی و دقت بهتری دارند جایگزین می‌شوند. روش‌های تشخیص مبتنی بر امضا دارای یک نقطه ضعف اساسی است. مشکل اساسی این روش عدم شناخت حملات جدید است که قبلاً در این سیستم تشخیص نفوذ وجود نداشته است و به روزرسانی دائم آن، یکی از نیازهای مهم آن‌ها برای بروزرسانی پایگاه‌داده امضاها است. در روش‌های اکتشافی فعالیت‌های مشکوک، نظیر وجود ترافیک غیرعادی در شبکه، ثبت کاراکترهای زده شده بر روی صفحه کلید در مرورگر، فعالیت‌های تحت شبکه‌ای که مشکوک هستند، دسترسی‌های غیر مجاز و.. . شناسایی می‌شوند. در صورتی که موارد گفته شده بیش از مقدار آستانه که قابل تغییر نیز است رخ دهد، حمله تشخیص داده شده و در غیر این شرایط عادی تلقی می‌شود. در این روش نیز تعیین مقدار آستانه کار ساده‌ای نیست و اگر به درستی تعیین نشود، تشخیص حملات دچار مشکل خواهد شد؛ زیرا امکان دارد یکسری از حملات را تشخیص نداده و یا تعدادی از عملیات عادی را حمله تشخیص دهد.

با ظهور و توسعه یادگیری ماشین و ابزارها و الگوریتم‌های مربوط به آن در حوزه‌های مختلف مانند داده‌کاوی، پردازش زبان‌های طبیعی، تشخیص گفتار و تبدیل گفتار به زبان، متخصصین امنیتی به فکر استفاده از ابزار و الگوریتم‌های موجود یادگیری ماشین افتادند و روش‌های متعددی را برای تشخیص بدافزار معرفی کردند. طی چند سال اخیر بحث پیرامون این روش‌ها بسیار رواج پیدا کرده و روش‌های مختلفی در مقالات علمی معتبر یافت می‌شود.

روزانه تعداد زیادی از حملات مختلف رخ می‌دهد. به همین دلیل تحلیل و بررسی حملات توسط نیروهای انسانی متخصص به دلیل کم بودن تعداد افراد متخصص نسبت به تعداد رو به افزایش این حملات و همچنین امکان وقوع خطای انسانی در تشخیص نفوذ امری خسته کننده و تقریبا ناممکن است. از این رو در سال‌های اخیر اقداماتی برای ماشینی کردن فرآیند تشخیص و جلوگیری از حملات انجام شد و نتیجه آن ساخت سیستم‌های تشخیص نفوذ متعددی شد که غالبا از روش‌های تشخیص امضا و یا روش‌های اکتشافی برای تشخیص بهره می‌برند.

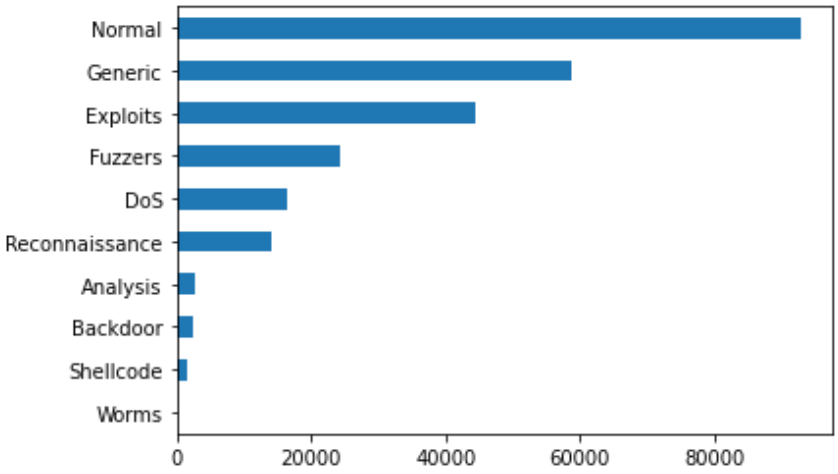
در این بخش، تکنیک‌ها/روش‌های مورد نیاز با مراحل روش‌شناختی برای انجام این پژوهش نشان داده شده است.

توضیحات مجموعه داده

بسته های شبکه خام مجموعه داده UNSW-NB 15 توسط ابزار IXIA PerfectStorm در آزمایشگاه Cyber Range در UNSW Canberra برای ایجاد ترکیبی از فعالیت های عادی مدرن واقعی و رفتارهای حمله مصنوعی معاصر ایجاد شده است. ابزار tcpdump برای ضبط ۱۰۰ گیگابایت از ترافیک خام استفاده شده. این مجموعه داده دارای ۹ نوع حمله است که عبارتند از Fuzzers ، Analysis، Backdoors، DoS، Exploits، Generic، Reconnaissance، Shellcode و Worms .از ابزارهای Argus و Bro-IDS استفاده می‌شود و دوازده الگوریتم برای تولید ۴۹ ویژگی با برچسب کلاس توسعه داده شده است. تعداد کل رکوردها ۲۵۴۰۰۴۴ است که در چهار فایل CSV به نام‌های UNSW-NB15_1.csv ، UNSW-NB15_2.csv، UNSW-NB15_3.csv و UNSW-NB15_4.csv ذخیره می‌شوند. پارتیشنی از این مجموعه داده به عنوان یک مجموعه آموزشی و مجموعه آزمایشی، به ترتیب UNSW_NB15_training-set.csv و UNSW_NB15_testing-set.csv پیکربندی شد. تعداد رکوردهای مجموعه آموزشی ۱۷۵۳۴۱ رکورد و مجموعه تست ۸۲۳۳۲ رکورد از انواع مختلف حمله و عادی می باشد.

جدول ۱- تعداد حملات چند کلاسه

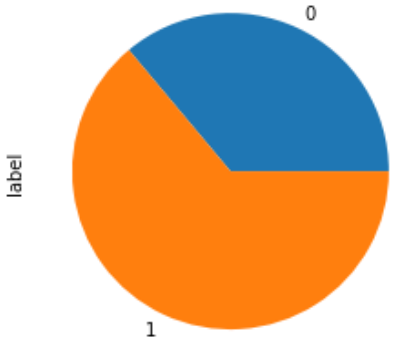
نوع حمله	تعداد
Normal	**93000**
Generic	**58871**
Exploits	**44525**
Fuzzers	**24246**
DoS	**16353**
Reconnaissance	**13987**
Analysis	**2677**
Backdoor	**2329**
Shellcode	**1511**
Worms	**174**



شکل ۲- تعداد حملات چند کلاسه

جدول ۲- تعداد حملات دو کلاسه

نوع	تعداد
Normal	**93000**
Attack	**164673**



شکل ۳- تعداد حملات دو کلاسه

همچنین ویژگی هایی که در این مجموعه داده وجود دارند به صورت جدول زیر است.

جدول ۳- ویژگی هایی که در این مجموعه داده

نام	توضیحات
srcip	آدرس IP منبع
sport	شماره پورت منبع

dstip	آدرس IP مقصد
dsport	شماره پورت مقصد
proto	پروتکل تراکنش
state	به حالت و پروتکل وابسته به آن نشان می دهد، به عنوان مثال ACC، CLO، CON، ECO، ECR، FIN، INT، MAS، PAR، REQ، RST، TST، TXD، URH، URN
dur	ضبط کل مدت زمان
sbytes	بایت تراکنش مبدأ به مقصد
dbytes	بایت تراکنش مقصد به منبع
sttl	زمان منبع تا مقصد
dttl	زمان مقصد به منبع
sloss	بسته های منبع مجددا ارسال یا حذف شده
dloss	بسته های مقصد مجددا ارسال شده یا از بین رفته
service	http, ftp, smtp, ssh, dns, ftp-data ,irc
Sload	بیت های منبع در ثانیه
Dload	بیت های مقصد در ثانیه
Spkts	تعداد بسته منبع تا مقصد
Dpkts	تعداد بسته مقصد تا منبع
swin	منبع TCP
dwin	مقصد TCP
stcpb	شماره توالی پایه TCP منبع
dtcpb	شماره توالی پایه TCP مقصد
smeansz	میانگین نحوه انتقال اندازه بسته توسط مبدا
dmeansz	میانگین نحوه انتقال اندازه بسته توسط مقصد
trans_depth	عمق خط لوله را در اتصال تراکنش درخواست/پاسخ http نشان می دهد
res_bdy_len	اندازه واقعی محتوای فشرده نشده داده های منتقل شده از سرویس http سرور.
Sjit	منبع جیتر (mSec)
Djit	مقصد جیتر (mSec)
Stime	رکورد زمان شروع
Ltime	رکورد زمان پایان
Sintpkt	زمان رسیدن بسته های داخلی منبع
Dintpkt	زمان رسیدن بسته های داخلی مقصد
tcprtt	زمان رفت و برگشت راه اندازی اتصال TCP
synack	زمان راه اندازی اتصال TCP

ackdat	زمان راه اندازی اتصال TCP، زمان بین بسته های SYN و ACK
is_sm_ips_ports	اگر آدرس های IP مبدا و مقصد برابر و شماره پورت برابر باشد، این متغیر مقدار ۱ را دریافت می کند و در غیر این صورت ۰
ct_state_ttl	شماره برای هر حالت با توجه به محدوده خاصی از مقادیر برای زمان مبدا/مقصد
ct_flw_http_mthd	تعداد جریان هایی که دارای روش هایی مانند دریافت و ارسال در سرویس http است.
is_ftp_login	اگر نشست ftp توسط کاربر و رمز عبور قابل دسترسی است، ۱ و در غیر این صورت ۰
ct_ftp_cmd	تعداد جریان هایی که در نشست ftp دستور دارند
ct_srv_src	تعداد اتصالاتی که مطابق آخرین زمان دارای آدرس سرویس و منبع یکسان در ۱۰۰ اتصال هستند
ct_srv_dst	تعداد اتصالاتی که دارای آدرس سرویس و مقصد یکسان در ۱۰۰ اتصال مطابق با آخرین زمان هستند
ct_dst_ltm	تعداد اتصالات همان آدرس منبع در ۱۰۰ اتصال طبق آخرین بار
ct_src_ ltm	تعداد اتصالات همان آدرس منبع در ۱۰۰ اتصال طبق آخرین بار
ct_src_dport_ltm	عدد صحیح، تعداد اتصالات همان آدرس مبدا و پورت مقصد در ۱۰۰ اتصال طبق آخرین بار
ct_dst_sport_ltm	تعداد اتصالات همان آدرس مقصد و پورت مبدا در ۱۰۰ اتصال طبق آخرین زمان
ct_dst_src_ltm	تعداد اتصالات همان مبدا و آدرس مقصد در ۱۰۰ اتصال طبق آخرین بار
attack_cat	نام هر دسته حمله. در این مجموعه داده، ۹ دسته وجود دارد. Fuzzers، Analysis، Backdoors، DoS، Exploits، Generic، Reconnaissance، Shellcode و Worms
Label	۰ برای عادی و ۱ برای سوابق حمله

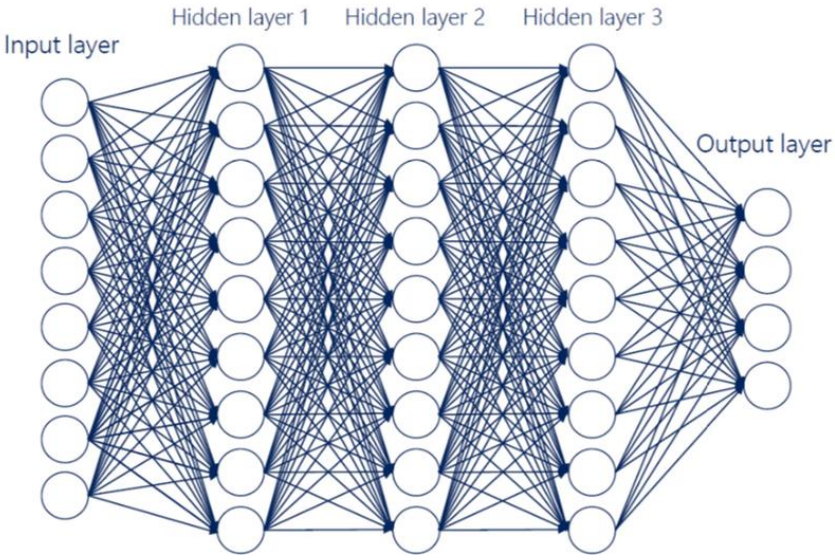
مدل های یادگیری عمیق پیشنهادی

ساختار و عمق مغز انسان بر یادگیری عمیق تأثیر گذاشت. شبکه یاد می گیرد که تابع ورودی را به خروجی نگاشت بدهد. فرآیند یادگیری به توانایی های مهندسی انتخاب ویژگی ها متکی نیست. بر اساس مجموعه ای از معیارها، ممکن است دنباله

ای از تکنیک های آماری برای تعیین اینکه آیا یک طبقه بندی بر اساس احتمال خطا صحیح است یا خیر، استفاده شود. در حوزه یادگیری عمیق، ما بر روی شبکه‌های عمیق با آموزش طبقه‌بندی در شبکه‌های سلسله مراتبی یادگیری بدون نظارت لایه‌های مختلف تمرکز می‌کنیم. مکانیسم های تشخیص نفوذ عمیق شبکه را می توان بسته به اجرای معماری و فناوری طبقه بندی کرد. مدل های مورد استفاده برای تجزیه و تحلیل در این بخش توضیح داده شده است. طبقه بندی کننده شبکه عصبی عمیق اولین مدل است. دومین مدل شبکه عصبی بازگشتی است. در نهایت مدل سوم از ترکیب دو مدل قبل بدست می‌آید. این مدل‌ها از دقت و از خطا در هر الگوریتم یادگیری عمیق و یادگیری ماشین برای محاسبه خروجی این مدل ها استفاده می کنند.

شبکه عصبی عمیق (DNN)

شبکه عصبی عمیق یک شبکه عصبی مصنوعی (ANN) با چندین لایه پنهان بین لایه ورودی و لایه خروجی یک شبکه عصبی عمیق (DNN) است. شبکه عصبی عمیق به دنبال روش ریاضی درستی است، رابطه میتواند خطی یا غیر خطی باشد تا ورودی را به خروجی تبدیل کند. DNN ها معمولاً شبکه های پیشخور هستند که در آنها داده ها از ورودی به لایه خروجی بدون حلقه بازگشتی جریان می یابد. DNN نقشه نورون مجازی را تولید می کند و مقادیر عددی دلخواه یا "وزن" را به اتصالات بین نورون ها اختصاص می دهد. ورودی و وزن‌ها ضرب می‌شوند و مقدار بین ۱ و ۰ برمی‌گردد. اگر شبکه دنباله‌ای را به درستی شناسایی نکرده باشد، الگوریتم وزن‌ها را تغییر می‌دهد. این به الگوریتم اجازه می دهد تا آن پارامترها را دستکاری کند تا زمانی که دستکاری ریاضی مناسب برای تکمیل پردازش داده ها تصمیم گیری شود[1].



شکل ۴- ساختار شبکه عصبی عمیق

پارامترهای مدل DNN در جدول ۴ توضیح داده شده است.

جدول ۴- پارامترهای شبکه عصبی عمیق

پارامتر	نوع / مقدار
تعداد لایه های پنهان	۴
تعداد نورون ها در هر لایه	۱۲۸-۶۴-۳۲-۱۶
نرخ یادگیری	۰.۰۰۲
تابع بهینه ساز	Adam
توابع فعال ساز	Tanh - Sigmoid
دوره	۵۰۰
اندازه دسته	۴۰۰۰

ساختار مدل ساخته شده برای دسته بندی ۱۰ گروهه به صورت زیر است .

```
Model_Attack_Category(
  (layers): Sequential(
    (0): Linear(in_features=41, out_features=128, bias=True)
    (1): Tanh()
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): Tanh()
    (4): Linear(in_features=64, out_features=32, bias=True)
    (5): Sigmoid()
    (6): Linear(in_features=32, out_features=16, bias=True)
    (7): Sigmoid()
    (8): Linear(in_features=16, out_features=10, bias=True)
  )
)
```

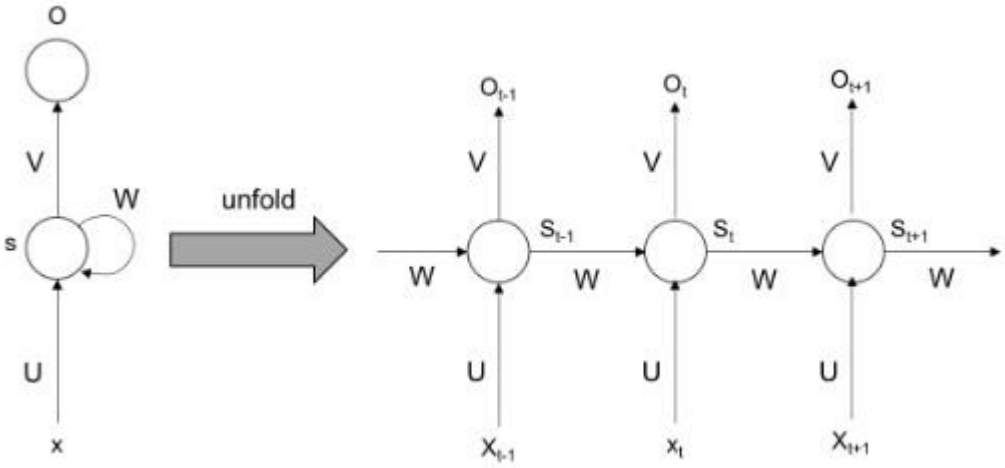
ساختار مدل ساخته شده برای دسته بندی ۲ گروهه به صورت زیر است .

```
Model_Label(
  (layers): Sequential(
    (0): Linear(in_features=41, out_features=128, bias=True)
    (1): Tanh()
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): Tanh()
    (4): Linear(in_features=64, out_features=32, bias=True)
    (5): Sigmoid()
    (6): Linear(in_features=32, out_features=16, bias=True)
    (7): Sigmoid()
    (8): Linear(in_features=16, out_features=2, bias=True)
  )
)
```

شبکه عصبی بازگشتی (RNN) با حافظه بلند مدت کوتاه مدت(LSTM)

یک شبکه‌ی عصبی بازگشتی کلاسی از شبکه‌های عصبی مصنوعی هستند که در آن اتصالات مابین گره‌هایی از یک گراف جهت‌دار در امتداد یک دنباله‌ی زمانی می‌باشند و سبب می‌شود تا الگوریتم بتواند رفتار پویای موقتی را به نمایش بگذارد. برخلاف شبکه‌های عصبی رو به جلو، شبکه‌های عصبی بازگشتی می‌توانند از وضعیت درونی خود برای پردازش دنباله‌ی

ورودی‌ها استفاده کنند. شبکه‌های عصبی بازگشتی، که قوی‌ترین و شناخته‌شده‌ترین آنها (LSTM) یا حافظه کوتاه مدت بلندمدت هستند، نوعی شبکه عصبی مصنوعی را منعکس می‌کنند که روندهای توالی داده‌ها را تشخیص می‌دهد.



شکل ۵- ساختار شبکه عصبی بازگشتی

پارامترهای مدل LSTM در جدول ۵ توضیح داده شده است.

جدول ۵- پارامترهای شبکه عصبی بازگشتی

پارامتر	نوع / مقدار
تعداد لایه های پشته	۳
دو طرفه	بله
اندازه لایه پنهان	۱۶
نرخ یادگیری	۰.۰۰۲
تابع بهینه ساز	Adam
دوره	۵۰۰
اندازه دسته	۱۰۰۰

ساختار مدل ساخته شده برای دسته بندی ۱۰ گروهه به صورت زیر است .

```
LSTM_Model_Attack_Category (
  (lstm): LSTM(41, 16, num_layers=3, batch_first=True, bidirectional=True)
  (fc): Linear(in_features=32, out_features=10, bias=True)
)
```

ساختار مدل ساخته شده برای دسته بندی ۲ گروهه به صورت زیر است .

```
LSTM_Model_Label(
  (lstm): LSTM(41, 16, num_layers=3, batch_first=True, bidirectional=True)
  (fc): Linear(in_features=32, out_features=2, bias=True)
)
```

ترکیب شبکه عصبی عمیق و شبکه عصبی بازگشتی

دو مدل قبل را با هم ترکیب کردیم و مدل جدیدی را ساختیم.

پارامترهای مدل جدید در جدول ۶ توضیح داده شده است.

جدول ۶- پارامترهای ترکیب شبکه عصبی عمیق و شبکه عصبی بازگشتی

پارامتر	نوع / مقدار
تعداد لایه های پشته	۲
دو طرفه	بله
dropout	۰.۵
اندازه لایه پنهان	۶۴
نرخ یادگیری	۰.۰۰۲
تابع بهینه ساز	Adam
دوره	۵۰۰
اندازه دسته	۲۰۰۰
تعداد لایه های پنهان	۴
تعداد نورون ها در هر لایه	۱۲۸-۶۴-۳۲
توابع فعال ساز	tanh-relu-sigmoid

ساختار مدل ساخته شده برای دسته بندی ۱۰ گروهه به صورت زیر است .

```
Model_Attack_Category (
  (lstm): LSTM(41, 64, num_layers=2, batch_first=True, dropout=0.5, bidirectional=True)
  (fc_layers): Sequential(
    (0): Linear(in_features=128, out_features=128, bias=True)
    (1): Tanh()
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): ReLU()
    (4): Linear(in_features=64, out_features=32, bias=True)
    (5): Sigmoid()
    (6): Linear(in_features=32, out_features=10, bias=True)
  )
)
```

ساختار مدل ساخته شده برای دسته بندی ۲ گروهه به صورت زیر است .

```
Model_label(
```

```
 (lstm): LSTM(41, 64, num_layers=2, batch_first=True, dropout=0.5, bidirectional=True)
 (fc_layers): Sequential(
   (0): Linear(in_features=128, out_features=128, bias=True)
   (1): Tanh()
   (2): Linear(in_features=128, out_features=64, bias=True)
   (3): ReLU()
   (4): Linear(in_features=64, out_features=32, bias=True)
   (5): Sigmoid()
   (6): Linear(in_features=32, out_features=2, bias=True)
 )
)
```

نتایج آزمایش مدل های یادگیری عمیق

در هر کدام از مدل‌ها، به دو صورت طبقه بندی داشتیم. هم به صورت چند کلاسه و هم به صورت دودویی. همچنین قبل از انجام عملیات آموزش مدل، پیش پردازش و نرمال‌سازی داده‌ها را انجام دادیم. و داده‌ها را به دو قسمت تقسیم کردیم که هفتاد درصد داده‌ها برای آموزش و سی درصد داده‌ها را به ارزیابی مدل اختصاص دادیم. معیارهای ارزیابی دقت، مقدار خطا و زمان آموزش به صورت یک دوره است. میانگین دقت اعتبارسنجی، میانگین خطا در آموزش محاسبه می‌شود تا نشان دهد که در مجموعه داده هیچ تناسب [1]یا کم‌برازشی[2] وجود ندارد. با توجه به تعداد دوره‌هایی که آموزش را تکمیل کرده‌اند، زمان آموزش را برای هر دوره دریافت می‌کنیم و از آن به عنوان معیار در ارزیابی ما استفاده می‌کنیم.

شبکه عصبی عمیق

ما مجموعا ۵۹ مدل با پارامترهای مختلف را در این شبکه ساختیم. نتایج مدل شبکه عصبی عمیق پیشنهادی ما برای هر دو طبقه‌بندی در جدول ۶ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی برای برچسب‌گذاری دودویی و برچسب‌گذاری چند کلاسه را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. شایان ذکر است، عملکرد شبکه عصبی مصنوعی در طبقه‌بندی باینری بهتر از طبقه‌بندی چند کلاسه بود. دقت برای مجموعه آزمایشی ۹۴٪ برای طبقه‌بندی دودویی و ۸۲.۴۶٪ برای دقت طبقه‌بندی چند کلاسه بود. در بخش آموزش خطا طبقه‌بندی دودویی۰.۱۱۳۷و ۰.۳۹۵۴ در طبقه‌بندی چند کلاسه بود. همچنین مدت زمان آموزش برای طبقه‌بندی دودیی ۹۸۶۳۷۵ میلی ثانیه و برای طبقه‌بندی چند کلاسه ۱۰۰۹۲۶۶ میلی ثانیه است.

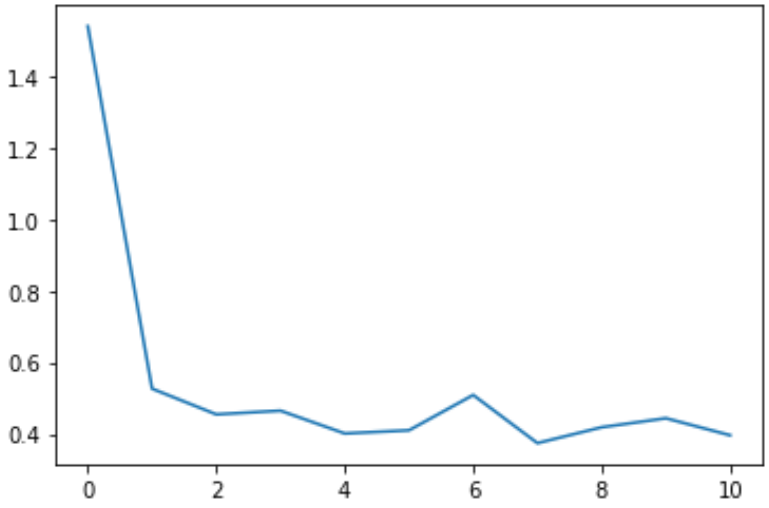
جدول ۷- نتایج مدل شبکه عصبی عمیق

معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
دقت آزمایش	۹۴٪	۸۲.۴۶٪
خطای آموزش	۰.۱۱۳۷	۰.۳۹۵۴
مدت زمان آموزش	۹۸۶۳۷۵ ms	۱۰۰۹۲۶۶ ms

[1] Overfitting

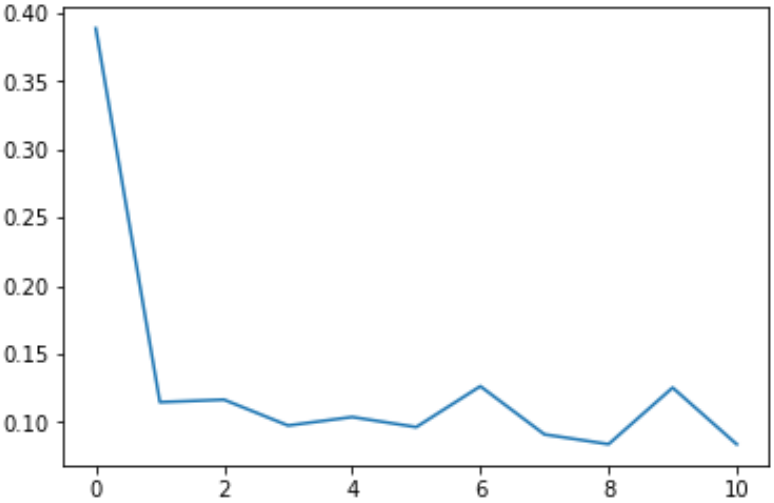
[2] underfitting

همچنین نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۶- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است .



شکل ۷- نمودار خطای طبقه‌بندی دو کلاسه

معیارهای ارزیابی Recall - Precision و F1-score

معیار Recall یا یادآوری

حداکثر مقدار این معیار یک ویا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که ما انتظار داشتیم پیش بینی شوند ولی برنامه پیش‌بینی نکرده‌است که به آن False Negative می‌گوییم نسبت به پیش بینی‌های درست یا True Positiveبیشتر باشد مقدار Recall کمتر خواهد شد.

فرمول محاسبه Recall

در فرمول زیر TP مخفف True Positive و FN مخفف False Negative است.

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{FN+TP}}$$

معیار Precision یا دقت

حداکثر مقدار این معیار یک ویا ۱۰۰ درصد و حداقل مقدار آن صفر است و هرچه مواردی که برنامه به **غلط پیش بینی کرده است** که به آن False Positive می‌گوییم نسبت به پیش بینی‌های درست یا True Positive بیشتر باشد مقدار Precisionکمتر خواهد شد.

فرمول محاسبه‌ی Precision

در فرمول زیر TP مخفف True Positive و FP مخفف False Positive است.

$$\textbf{Precision} = \frac{TP}{TP+FP}$$

معیارf1-score

زمانی که می‌خواهید معیار ارزیابی شما میانگینی از دو مورد قبلی باشد یعنی همان Recall یا Precision می‌توانید از میانگین هارمونیک این دو معیار استفاده کنید که به آن **معیار f1-score** می‌گویند.

$$\textbf{F1-score} = 2 * \frac{\textbf{Precision} * \textbf{Recall}}{\textbf{Precision+ Recall}}$$

معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه به صورت زیر است.

جدول ۸- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه

کلاس	precision	recall	f1-score
۰	0.75	0.06	**0.12**
۱	0.74	0.06	**0.11**
۲	0.50	0.07	**0.13**
۳	0.61	0.92	**0.73**
۴	0.66	0.56	**0.61**
۵	0.99	0.98	**0.99**
۶	0.91	0.93	**0.92**

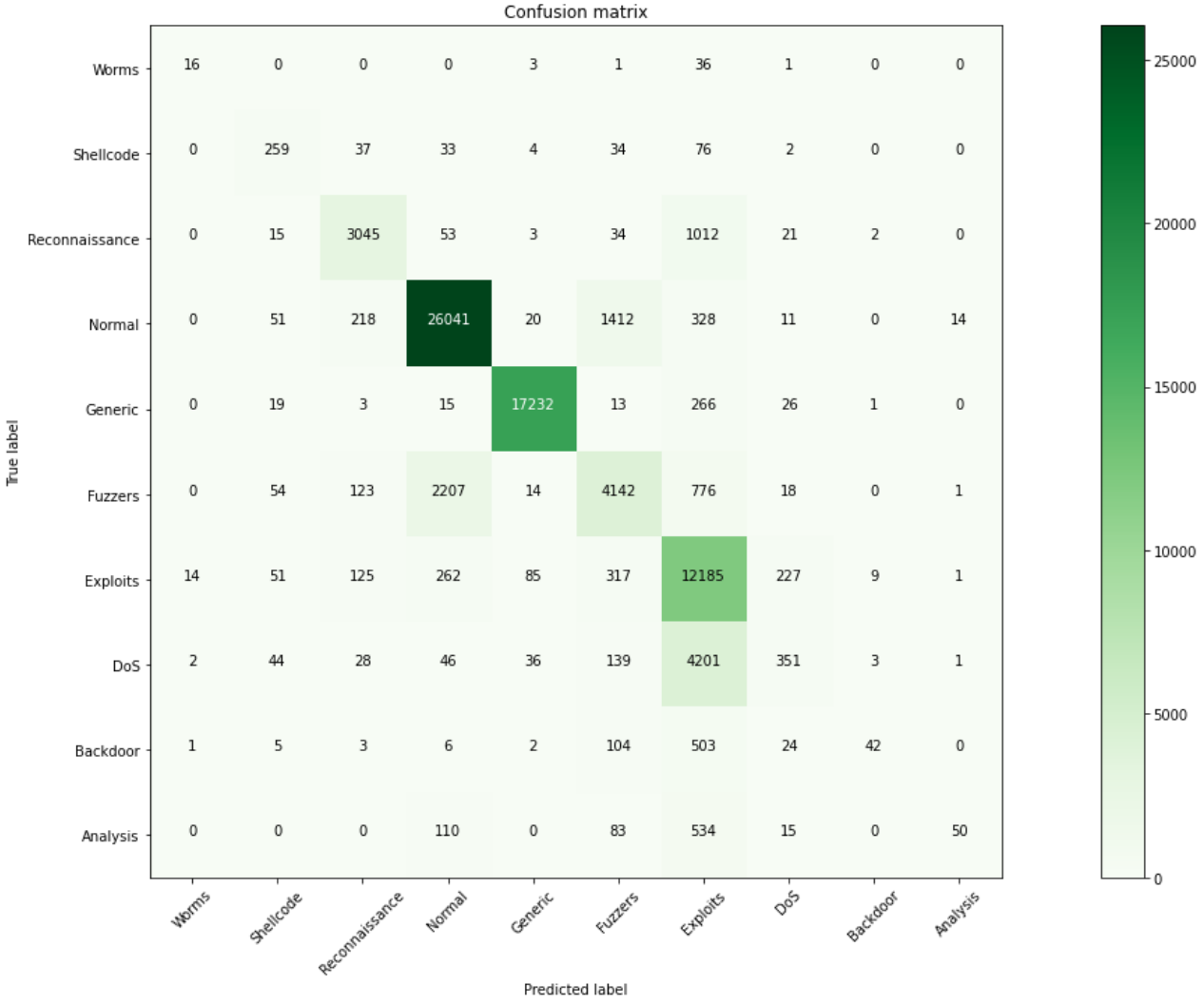
۷	0.85	0.73	**0.78**
۸	0.52	0.58	**0.55**
۹	0.48	0.28	**0.36**

همچنین معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه به صورت زیر است.

جدول ۹- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه

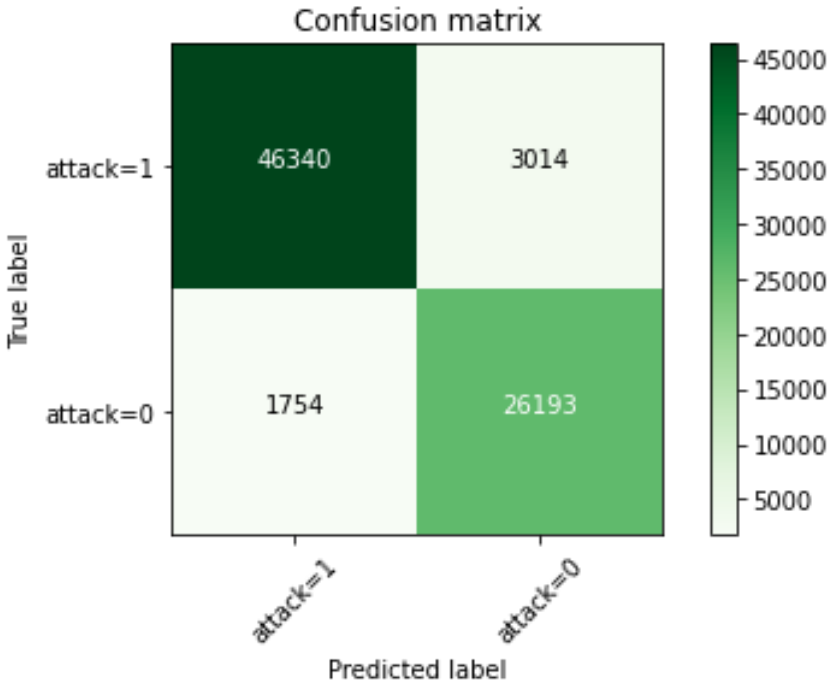
کلاس	precision	recall	f1-score
۰	0.90	0.94	**0.92**
۱	0.96	0.94	**0.95**

ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه به صورت زیر است



شکل ۸- ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه

ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۹- ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه

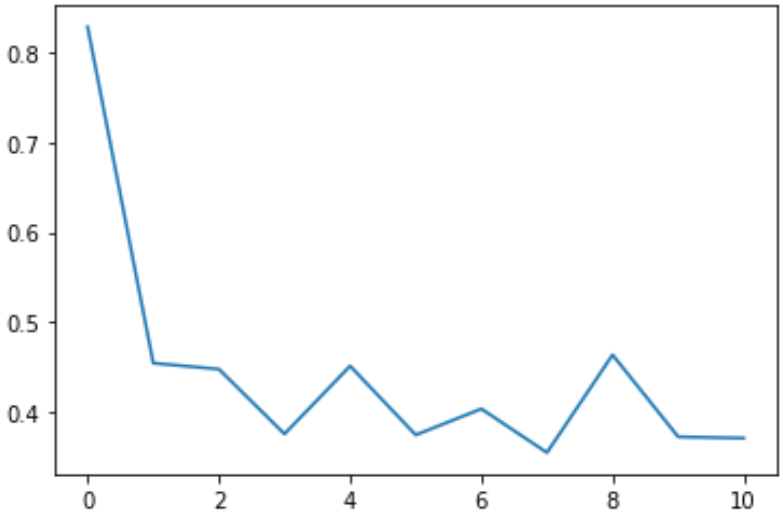
شبکه عصبی بازگشتی

ما مجموعا ۶۷ مدل با پارامترهای مختلف را در این شبکه ساختیم. نتایج مدل شبکه عصبی بازگشتی پیشنهادی ما برای هر دو طبقه‌بندی در جدول ۱۰ نشان داده شده است. جدول نتایج شبکه عصبی مصنوعی LSTM برای برچسب‌گذاری دودویی و برچسب‌گذاری چند کلاسه را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی۹۳.۸۹ ٪ برای طبقه‌بندی دودویی و ۸۲.۳۳٪ برای دقت طبقه‌بندی چند کلاسه بود. در بخش آموزش خطا طبقه‌بندی دودویی۰.۰۸۴۶ و ۰.۳۵۹۳ در طبقه‌بندی چند کلاسه بود. همچنین مدت زمان آموزش برای طبقه‌بندی دودیی ۱۲۸۹۲۹۱میلی ثانیه و برای طبقه‌بندی چند کلاسه ۱۲۷۶۳۰۲ میلی ثانیه است.

جدول ۱۰- نتایج شبکه عصبی بازگشتی

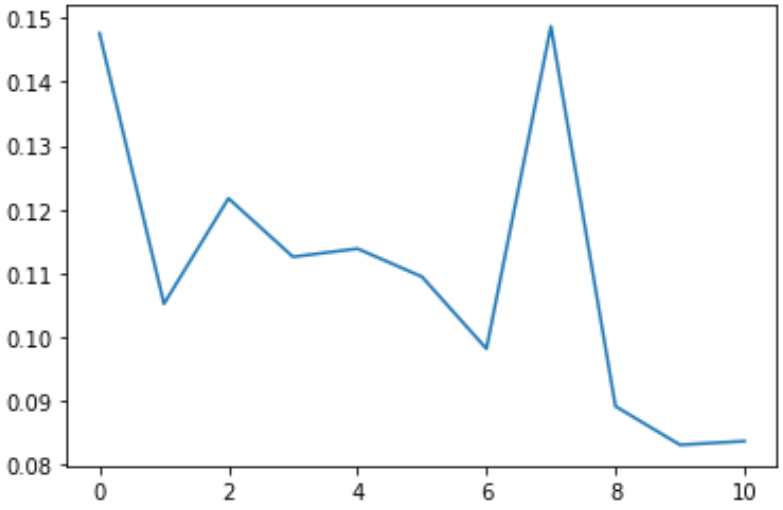
معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
دقت آزمایش	٪ ۹۳.۸۹	۸۲.۳۳٪
خطای آموزش	۰.۰۸۴۶	۰.۳۵۹۳
مدت زمان آموزش	۱۲۸۹۲۹۱ms	۱۲۷۶۳۰۲ ms

همچنین نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۱۱- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است .



شکل ۱۰- نمودار خطای طبقه‌بندی دو کلاسه

معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه به صورت زیر است.

جدول ۱۱- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه

کلاس	precision	recall	f1-score
۰	۰.۸۱	۰.۰۶	**۰.۱۲**
۱	0.76	0.08	**0.15**
۲	0.42	0.17	**0.24**

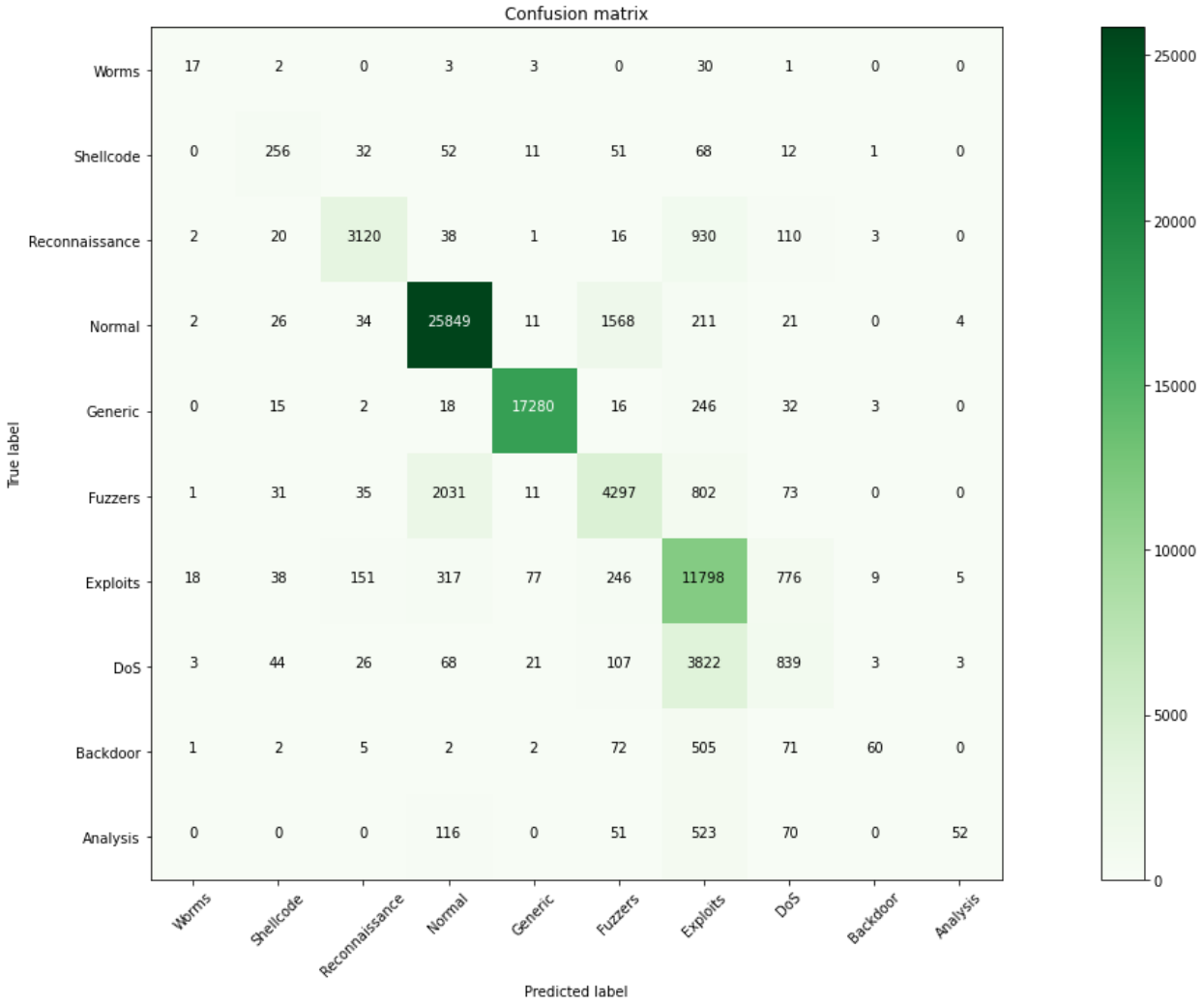
۳	0.62	0.88	**0.73**
۴	0.67	0.59	**0.63**
۵	0.99	0.98	**0.99**
۶	0.91	0.93	**0.92**
۷	0.92	0.74	**0.82**
۸	0.59	0.53	**0.56**
۹	0.39	0.3	**0.34**

همچنین معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه به صورت زیر است.

جدول ۱۲- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه

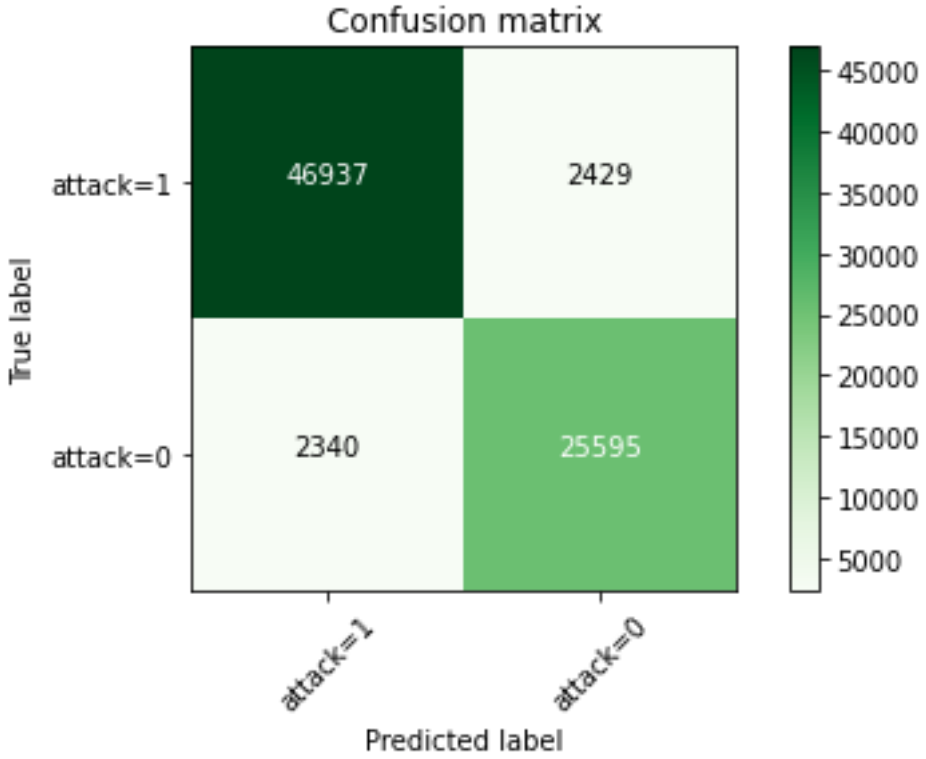
کلاس	precision	recall	f1-score
۰	0.91	0.92	**0.91**
۱	0.95	0.95	**0.95**

ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه به صورت زیر است



شکل ۱۲- ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه

ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۱۳- ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه

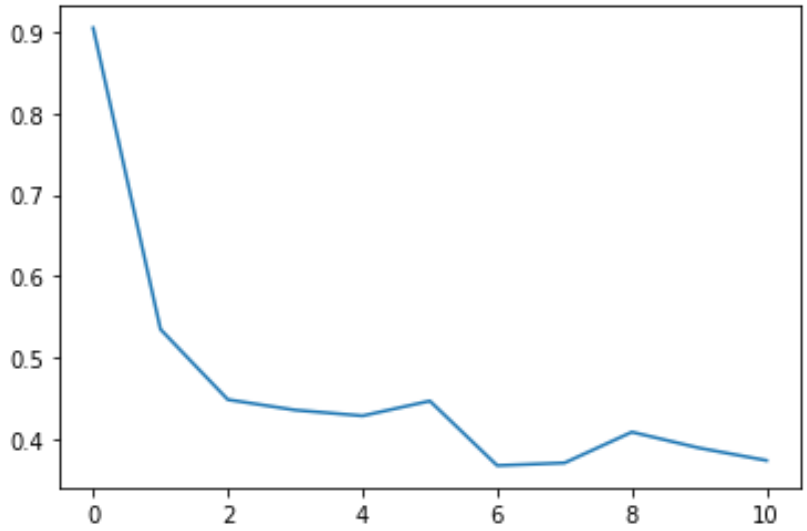
ترکیب دو شبکه عصبی عمیق و شبکه عصبی بازگشتی

ما مجموعا ۲۰ مدل با پارامترهای مختلف را در این شبکه ساختیم. نتایج مدل پیشنهادی ما برای هر دو طبقه‌بندی در جدول ۱۳ نشان داده شده است. جدول نتایج برای برچسب‌گذاری دودویی و برچسب‌گذاری چند کلاسه را از نظر دقت، خطا و زمان آموزش برای هر دوره نشان می‌دهد. دقت برای مجموعه آزمایشی۹۳.۹۷ ٪ برای طبقه‌بندی دودویی و ۸۲.۴۹٪ برای دقت طبقه‌بندی چند کلاسه بود. در بخش آموزش خطا طبقه‌بندی دودویی۰.۱۳۵۵ و ۰.۳۴۸۲ در طبقه‌بندی چند کلاسه بود. همچنین مدت زمان آموزش برای طبقه‌بندی دودیی ۱۱۳۰۲۳۱ میلی ثانیه و برای طبقه‌بندی چند کلاسه ۱۱۲۵۱۰۹ میلی ثانیه است.

جدول ۱۳- نتایج مدل ترکیب دو شبکه عصبی عمیق و شبکه عصبی بازگشتی

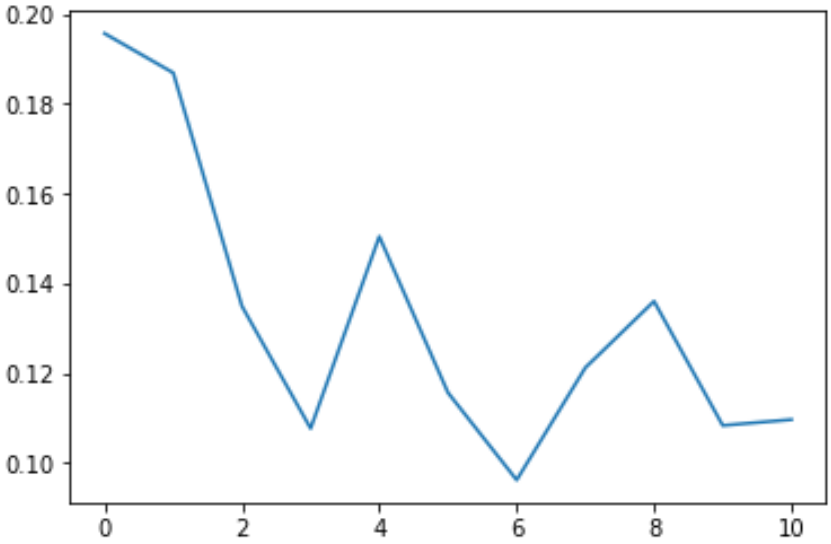
معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
دقت آزمایش	٪ ۹۳.۹۷	۸۲.۴۹٪
خطای آموزش	۰.۱۳۵۵	۰.۳۴۸۲
مدت زمان آموزش	۱۱۳۰۲۳۱ ms	۱۱۲۵۱۰۹ ms

همچنین نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۱۴- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است .



شکل ۱۵- نمودار خطای طبقه‌بندی دو کلاسه

معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه به صورت زیر است.

جدول ۱۴- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی چند کلاسه

کلاس	precision	recall	f1-score
۰	۰.۷۶	۰.۰۵	۰.۱۰
۱	0.63	0.07	**0.13**

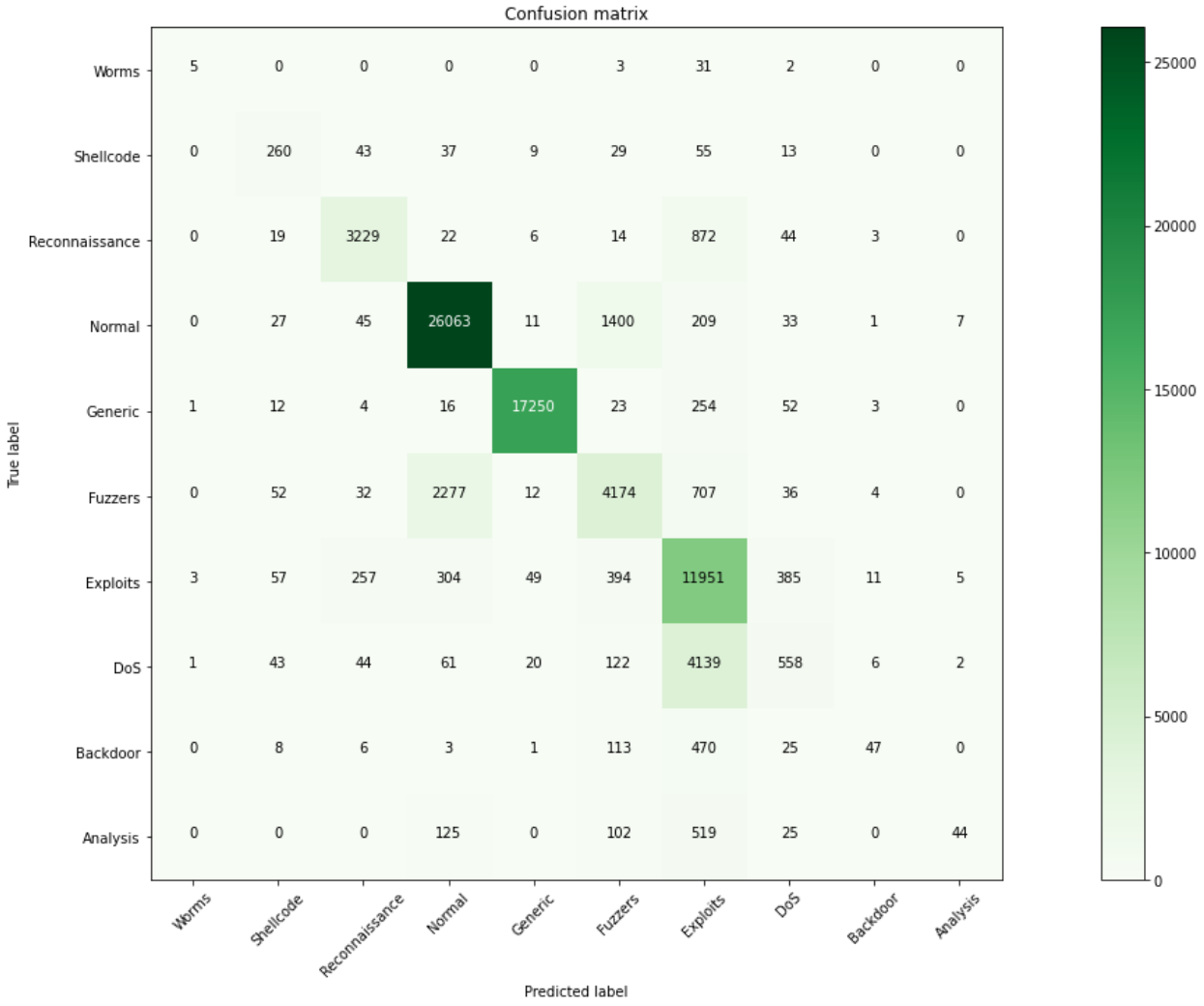
۲	0.48	0.11	**0.18**
۳	0.62	0.89	**0.73**
۴	0.65	0.57	**0.61**
۵	0.99	0.98	**0.99**
۶	0.90	0.94	**0.92**
۷	0.88	0.77	**0.82**
۸	0.54	0.58	**0.56**
۹	0.5	0.12	**0.2**

همچنین معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه به صورت زیر است.

جدول ۱۵- معیارهای ارزیابی Recall - Precision و F1-score برای طبقه‌بندی دو کلاسه

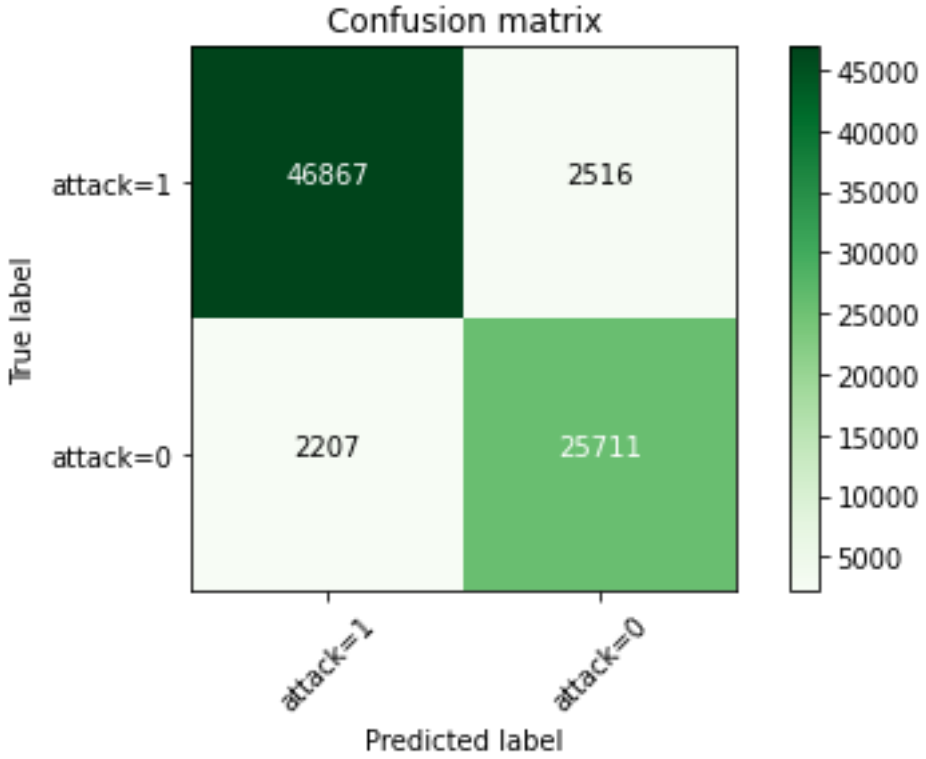
کلاس	precision	recall	f1-score
۰	0.91	0.92	**0.92**
۱	0.96	0.95	**0.95**

ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه به صورت زیر است



شکل ۱۶- ماتریس درهم ریختگی برای طبقه‌بندی چند کلاسه

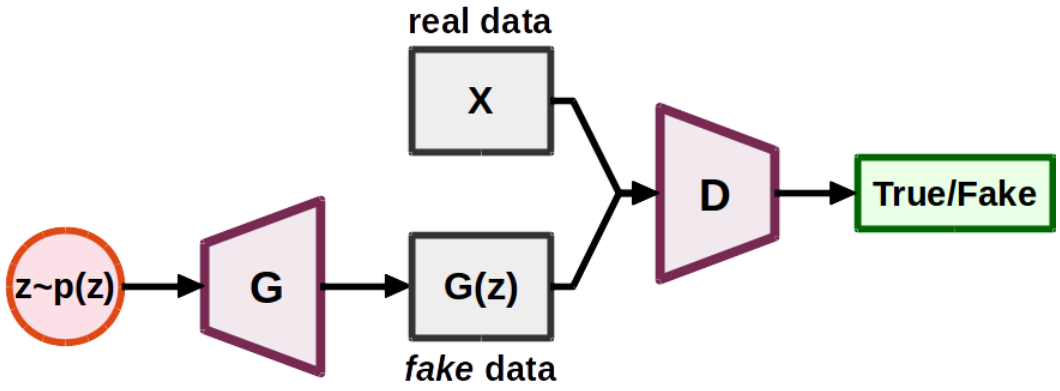
ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۱۷- ماتریس درهم ریختگی برای طبقه‌بندی دو کلاسه

حمله به مدل با استفاده از شبکه های عصبی GAN

ما در مجموع ۱۴۶ مدل از انواع شبکه های عصبی ساختیم و نتایج ۳ مدل با بهترین نتایج را در مقاله ذکر کردیم. در گام بعد به ۳ مدلی که بهترین نتایج را داشتند با استفاده از شبکه های عصبی GAN حمله کردیم تا دقت مدل خود را ارزیابی کنیم. روال کار در شبکه های عصبی GAN به این صورت است که یک بخش Generator برای تولید داده های جعلی دارد و یک بخش به نام Discriminator دارد برای تشخیص داده های جعلی از واقعی. سه مدلی که بهترین نتایج را داشتند را هر بار به عنوان Discriminator در نظر میگیریم و با تولید داده های جعلی از طریق Generator به مدل حمله میکنیم.



شکل ۱۷- شبکه های عصبی GAN

پارامترهای مدل GAN برای حمله به شبکه عصبی عمیق

جدول ۱۶- پارامترهای مدل GAN برای حمله به شبکه عصبی عمیق

پارامتر	نوع / مقدار
اندازه ورودی Generator	۲۰
نرخ یادگیری	3e-4
تابع بهینه ساز	Adam
تابع فعال ساز	LeakyReLU
تابع هدررفت	BCELoss
دوره	۵۰
اندازه دسته	۴۰۰۰

ساختار مدل ساخته شده Generator

```
Generator(
  (gen): Sequential(
    (0): Linear(in_features=20, out_features=16, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
    (2): Linear(in_features=16, out_features=32, bias=True)
    (3): LeakyReLU(negative_slope=0.01)
    (4): Linear(in_features=32, out_features=64, bias=True)
    (5): LeakyReLU(negative_slope=0.01)
    (6): Linear(in_features=64, out_features=128, bias=True)
    (7): LeakyReLU(negative_slope=0.01)
    (8): Linear(in_features=128, out_features=41, bias=True)
  )
)
```

ساختار مدل ساخته شده Discriminator

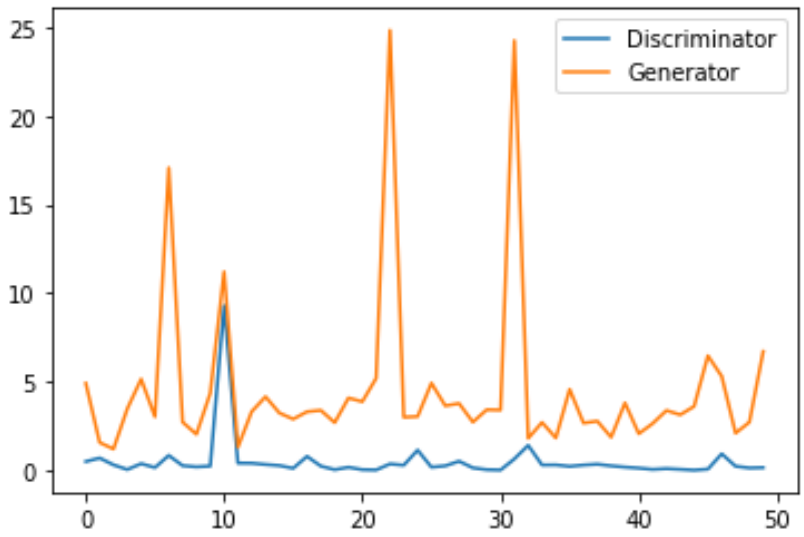
```
Discriminator(
  (disc): Sequential(
    (0): Linear(in_features=41, out_features=128, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): LeakyReLU(negative_slope=0.01)
    (4): Linear(in_features=64, out_features=32, bias=True)
    (5): LeakyReLU(negative_slope=0.01)
    (6): Linear(in_features=32, out_features=16, bias=True)
    (7): LeakyReLU(negative_slope=0.01)
    (8): Linear(in_features=16, out_features=1, bias=True)
    (9): Sigmoid()
  )
)
```

نتایج حاصله از اجرای این کار به صورت جدول زیر است.

جدول ۱۷- نتایج مدل GAN برای حمله به شبکه عصبی عمیق

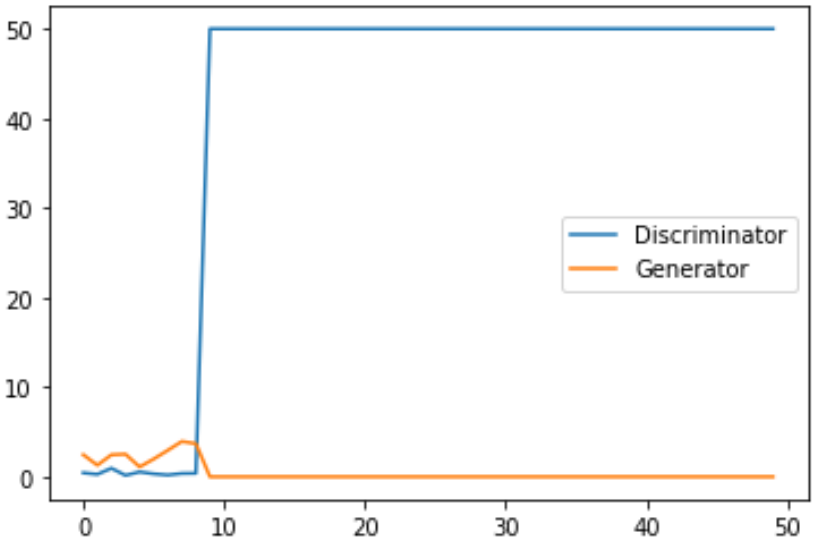
معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
خطای Generator	۰.۰۰۰۰	۲.۷۴۴۲
خطای Discriminator	۵۰.۰۰۰۰	۰.۱۵۱۰

نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۱۸- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۲۰- نمودار خطای طبقه‌بندی دو کلاسه

پارامترهای مدل GAN برای حمله به شبکه عصبی بازگشتی

جدول ۱۸- پارامترهای مدل GAN برای حمله به شبکه عصبی بازگشتی

پارامتر	نوع / مقدار
اندازه ورودی Generator	۲۰
نرخ یادگیری	3e-6
تابع بهینه ساز	Adam
تابع فعال ساز	LeakyReLU
تابع هدررفت	BCELoss
دوره	۵۰
اندازه دسته	۴۰۰۰

ساختار مدل ساخته شده Generator

```
Generator(
  (gen): Sequential(
    (0): Linear(in_features=20, out_features=16, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
    (2): Linear(in_features=16, out_features=32, bias=True)
    (3): LeakyReLU(negative_slope=0.01)
    (4): Linear(in_features=32, out_features=64, bias=True)
    (5): LeakyReLU(negative_slope=0.01)
    (6): Linear(in_features=64, out_features=128, bias=True)
    (7): LeakyReLU(negative_slope=0.01)
    (8): Linear(in_features=128, out_features=41, bias=True)
```

```
  )
)
```

ساختار مدل ساخته شده Discriminator

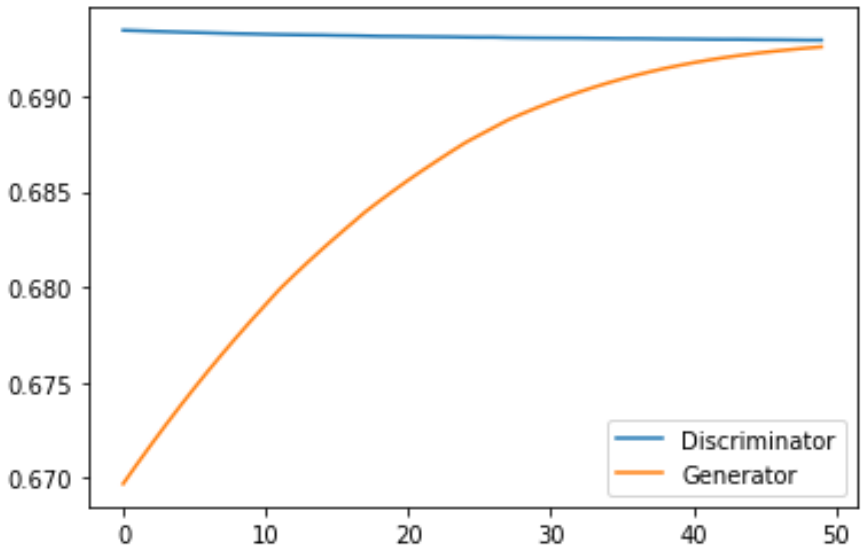
```
Discriminator(
  (lstm): LSTM(41, 16, num_layers=3, batch_first=True, bidirectional=True)
  (fc_layers): Sequential(
    (0): LeakyReLU(negative_slope=0.01)
    (1): Linear(in_features=32, out_features=128, bias=True)
    (2): LeakyReLU(negative_slope=0.01)
    (3): Linear(in_features=128, out_features=64, bias=True)
    (4): LeakyReLU(negative_slope=0.01)
    (5): Linear(in_features=64, out_features=1, bias=True)
    (6): Sigmoid()
  )
)
```

نتایج حاصله از اجرای این کار به صورت جدول زیر است.

جدول ۱۹- نتایج مدل GAN برای حمله به شبکه عصبی بازگشتی

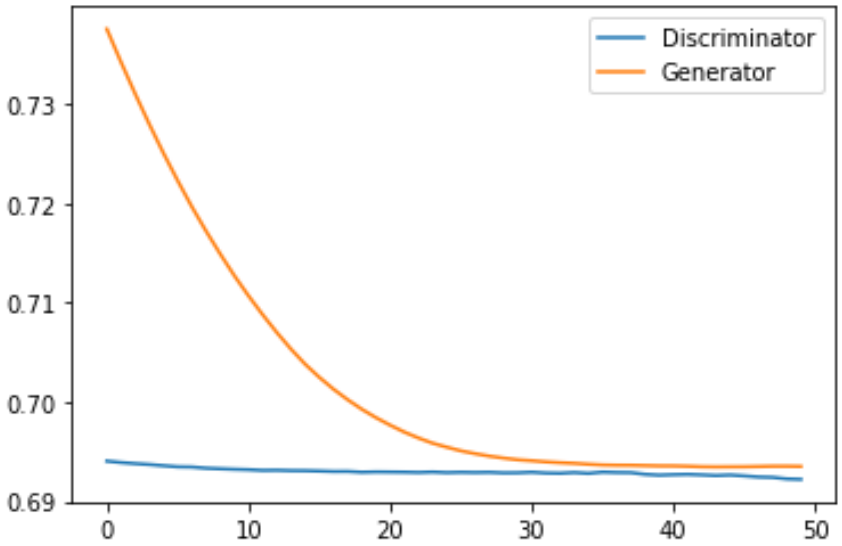
معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
خطای Generator	۰.۶۹۳۵	۰.۶۹۲۵
خطای Discriminator	۰.۶۹۲۴	۰.۶۹۲۹

نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۱۹- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۲۲- نمودار خطای طبقه‌بندی دو کلاسه

پارامترهای مدل GAN برای حمله به ترکیب دو شبکه عصبی عمیق و شبکه عصبی بازگشتی

جدول ۲۰- پارامترهای مدل GAN برای حمله به ترکیب دو شبکه عصبی عمیق و شبکه عصبی بازگشتی

پارامتر	نوع / مقدار
اندازه ورودی Generator	۲۰
نرخ یادگیری	3e-6
تابع بهینه ساز	Adam
تابع فعال ساز	LeakyReLU
تابع هدررفت	BCELoss
دوره	۵۰
اندازه دسته	۴۰۰۰

ساختار مدل ساخته شده Generator

```
Generator(
  (gen): Sequential(
    (0): Linear(in_features=20, out_features=16, bias=True)
    (1): LeakyReLU(negative_slope=0.01)
    (2): Linear(in_features=16, out_features=32, bias=True)
    (3): LeakyReLU(negative_slope=0.01)
    (4): Linear(in_features=32, out_features=64, bias=True)
    (5): LeakyReLU(negative_slope=0.01)
```

```
    (6): Linear(in_features=64, out_features=128, bias=True)
    (7): LeakyReLU(negative_slope=0.01)
    (8): Linear(in_features=128, out_features=41, bias=True)
  )
)
```

ساختار مدل ساخته شده Discriminator

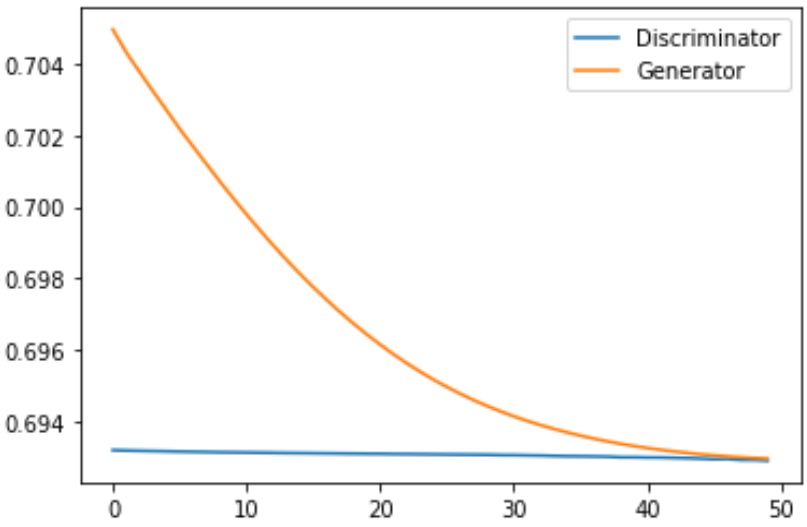
```
Discriminator(
  (lstm): LSTM(41, 16, num_layers=3, batch_first=True, bidirectional=True)
  (fc_layers): Sequential(
    (0): LeakyReLU(negative_slope=0.01)
    (1): Linear(in_features=32, out_features=128, bias=True)
    (2): LeakyReLU(negative_slope=0.01)
    (3): Linear(in_features=128, out_features=64, bias=True)
    (4): LeakyReLU(negative_slope=0.01)
    (5): Linear(in_features=64, out_features=32, bias=True)
    (6): LeakyReLU(negative_slope=0.01)
    (7): Linear(in_features=32, out_features=1, bias=True)
    (8): Sigmoid()
  )
)
```

نتایج حاصله از اجرای این کار به صورت جدول زیر است.

جدول ۲۱- نتایج مدل GAN برای حمله به ترکیب دو شبکه عصبی عمیق و شبکه عصبی بازگشتی

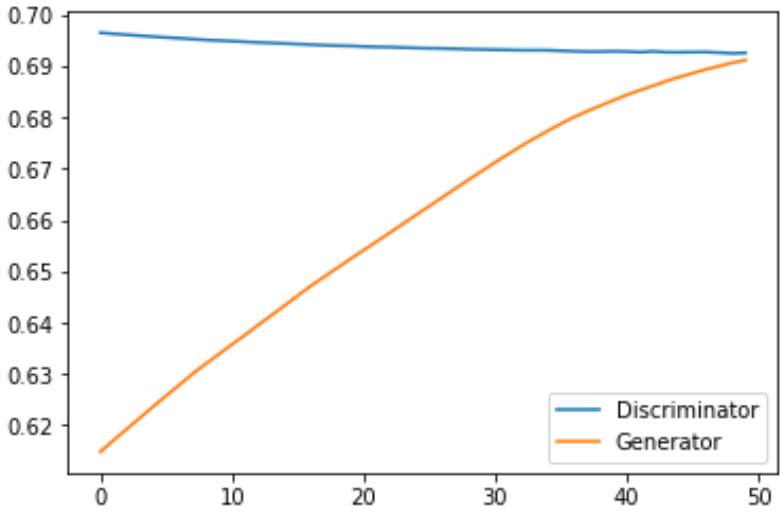
معیار	طبقه‌بندی دودویی	طبقه‌بندی چند کلاسه
خطای Generator	۰.۶۹۰۶	۰.۶۹۳۰
خطای Discriminator	۰.۶۹۲۶	۰.۶۹۲۹

نمودار خطای طبقه‌بندی چند کلاسه به صورت زیر است .



شکل ۲۰- نمودار خطای طبقه‌بندی چند کلاسه

و نمودار خطای طبقه‌بندی دو کلاسه به صورت زیر است.



شکل ۲۱- نمودار خطای طبقه‌بندی دو کلاسه

نتیجه گیری

این تحقیق مدل‌های یادگیری عمیق مبتنی بر ANN، DNN و RNN را به عنوان یک سیستم تشخیص نفوذ پیشنهاد می‌کند که یک سیستم تشخیص نفوذ مشارکتی جدید برای تشخیص فعالیت‌های نفوذی در محیط‌های محاسباتی است. این تحقیق شامل پیش پردازش داده UNSW-NB15 است تا با مدل های یادگیری عمیق برای شناسایی الگوهای غیرعادی استفاده شود. همچنین پس از ساخت مدل‌ها با دقت‌های مطلوب، به مدل خود حمله کردیم تا آن را ارزیابی کنیم.

منابع

[1] Aleesa, A.; Zaidan, B.; Zaidan, A.; and Sahar, N.M. (2020). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy,challenges, motivations, recommendations, substantial analysis and future directions. Neural Computing and Applications, 32(14), 9827-9858.