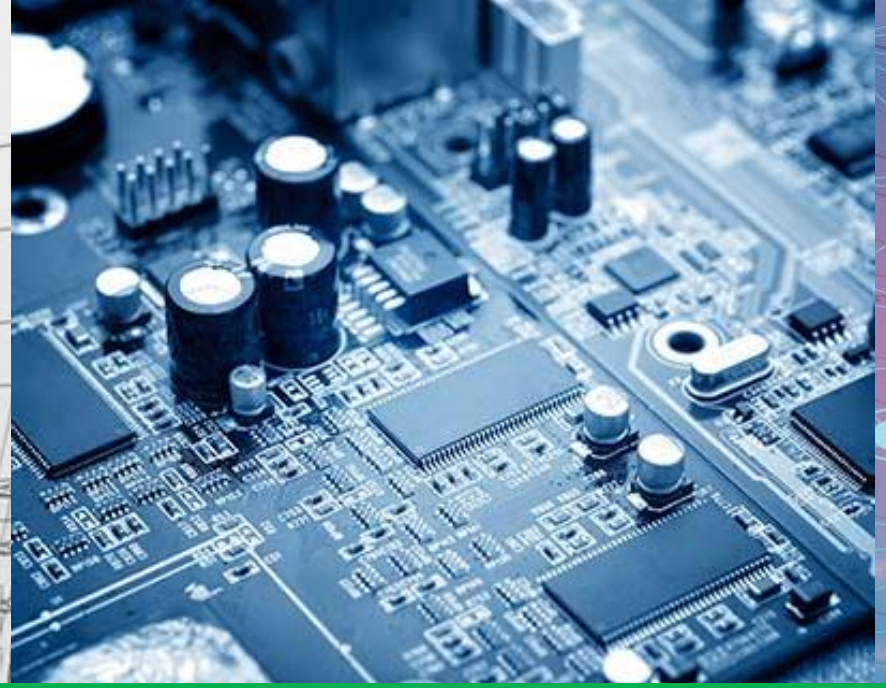
پروژه نهایی درس شناسایی الگو

[لینک]

DoS attack detection model of smart grid based on machine learning method

Wang Zhe, Cheng Wei, Li Chunlin*

نام استاد: دکتر یزدی
نام دانشجو : حمزه قائدی
شماره دانشجویی: ۹۸۳۱۴۱۹
زمستان ۹۹



شبکه هوشمند = فناوری اطلاعات و ارتباطات + شبکه سنتی

مقدمه

- مقایسه با شبکه سنتی
- حملات سایبری در شبکه هوشمند
- راهکارهای ارائه شده تاکنون
- راهکار مقاله

۱

در سراسر مسیر توزیع، سنسورها و پردازنده هایی برای ثبت و پردازش اطلاعات محلی شبکه(ولتار و جریان نودها خرابی اجزا و... تعبیه شده است

۲

نود های شبکه (مولد، مصرف کننده و..) اطلاعات پردازش شده را بین یدکدیگر مخابره میکنند، این امر، مصرف، عیب یابی و نگه داری را به صورت هوشمندانه امکان پذیر می نماید

۳

در شبکه سنتی، هر مرکز پایش و توزیع قدرت، بر بخش وسیعی از شبکه نظارت دارد (معماری متمرکز)

۴

در شبکه هوشمند، هر نود میتواند به اطلاعات سایر نود ها دسترسی داشته باشد. لذا میتواند بالقوه بر بخشی از شبکه نظارت کند بنابراین وظیفه پایش و کنترل شبکه بین نودها توزیع میشود (معماری غیر متمرکز)

مقدمه

مقایسه با شبکه سنتی

۱

منابع تجدیدپذیر[1]، عموما ناپایدار اند همین امر بهره برداری از این نوع منابع را در شبکه سنتی دشوار نموده است.

شبکه هوشمند، کنترل منابع تجدیدپذیر را تسهیل میکند

۲

معماری نامتمرکز[2]، امکان حذف یا افزودن نود به شبکه را تسهیل میکند.

(شبکه هوشمند منعطف[3] تر است)

۳

در شبکه سنتی، نیروگاه های برق در فواصل دور از مصرف کنند گان قرار دارند و تجهیزات انتقال فشار قوی نیز گرانقیمت اند در مقابل، تولید کنندگان در شبکه هوشمند کوچک بوده، آلودگی نداشته و نزدیک به مصرف کنندگان قرار میگیرند

1. Renewable sources
2. Decentralized
3. Flexibility
4. Passive
5. Active

شبکه سنتی	شبکه هوشمند
تعداد کمی نیروگاه بزرگ	تعداد زیادی مولد کوچک
شبکه انتقال متمرکز	شبکه انتقال نامتمرکز
متکی بر خطوط انتقال بزرگ	متشکل از خطوط انتقال کوچک به همراه جبرانساز های محلی
انتقال توان تنها از مولد به مصرف کننده (یک طرفه)	انتقال توان دو طرفه
فقط مصرف[4]	هم تولید هم مصرف[5]

مقدمه

حملات سایبری در شبکه هوشمند

✓ **مخابره اطلاعات، مهمترین وجه تمییز شبکه هوشمند نسبت به شبکه سنتی است**

✓ **اینترنت بستری مناسب برای مخابره اطلاعات در شبکه هوشمند است**

شبکه هوشمند، در برابر انواع حملات سایبری، آسیب پذیر است

مثال: در ششم ژانویه سال ۲۰۱۶، شبکه قدرت اکراین مورد حمله قرار گرفت و برق بخش وسیعی از اکراین قطع شد. این اولین حمله موفق به شبکه قدرت یک کشور بود.

راهکارهای مقابله

✓ طراحی شبکه به گونه ایکه حداقل تعداد نقاط شکست[1] را داشته باشد

✓ **استفاده از روشهای تشخیص حمله**

✓ برنامه نویسی صحیح اجزای شبکه

1. Failure point
2. Band width
3. Buffer overflow
4. Router
5. Packet loss

حمله DoS

✓ با اشغال منابع شبکه[2] (نظیر پهنای باند)، شبکه را به طور موقت مختل میکند (یا سیستم را بسیار کند میکند) **و بر دو نوع است:**

۱ Flooding

- سرریز بافر[3]: مرسوم ترین روش، تعداد زیاد پکت های ارسالی باعث سریز بافر در روترها[4] شده و اشغال پهنای باند و پکت لاس[5] خواهیم داشت
- Smurf: استفاده از اشتباهات در پیکربندی شبکه
- SYN Flood
- Neptune

و...

۲ Crashing

- ارسال پکتهایی مشخص به منظور استفاده از باگ های سیستم

چرا DoS مهم است ؟

✓ ساده است و به راحتی قابل پیاده سازیست و لذا مرسوم است

✓ نوع توزیع شده آن(DDoS) بسیار موثر است

✓ روش (Crashing) که وابسته به باگهای برنامه نویسی است، اگرچه پیچیده تر است اما مخربتر بود و تشخیص آن دشوار است

مقدمه

پیشینه پژوهش/
راهکار مقاله

- استفاده از تحلیل بازی برپایه زنجیره مارکوف[1] به منظور تشخیص حملات تزیق داده معیوب[2] (پژوهشگر: Eunsuk Kang)
  روش فوق نمیتواند حملات سایبری مرسوم شبکه قدرت (نظیر DoS) را تشخیص دهد
  **) راه حل مقاله: مدل ارائه شده به کمک یادگیری ماشین، منحصرا حملات DoS را تشخیص میدهد.

- تشخیص حمله بر اساس روش Alarm Data Fusion (نویسنده: Yanan Sun)
  این روش، تنها غیر عادی بودن[3] وضعیت خط را تشخیص میدهد و قادر به تعیین محل دقیق نود معیوب نمی باشد
  **) راه حل مقاله: مدل ارائه شده به سادگی روی میکروکنترلرهای ارزان قیمت مرسوم قابل پیاده سازی است و میتوان آنرا در قالب ماژول در نقاط مخلف شبکه تعبیه کرد.

اهداف مدل مطرح شده در مقاله

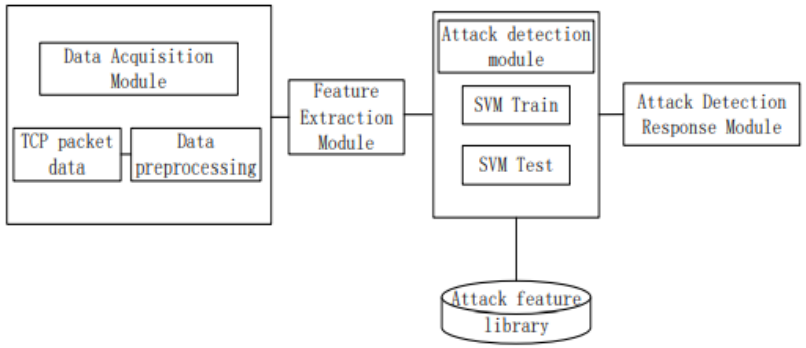


Fig. 3 Attack detection model

- ✓ ارازنی : تا تعبیه تعداد زیاد آن در نقاط مختلف شبکه مقرون به صرفه باشد
- ✓ پیچدگی محاسباتی کم: تا روی میکروکنترلرهای ارزان قیمت موجود قابل پیاده سازی باشد
- ✓ سرعت تشخیص بالا : تا حضور آن بازدهی شبکه را چندان کاهش ندهد

((ماژول طراحی شده یک سامانه بلادرنگ[4] است و باید تمام پکتهای دریافتی[5] را بررسی کند و موارد معیوب را تشخیص دهد))

- ✓ دقت بالا

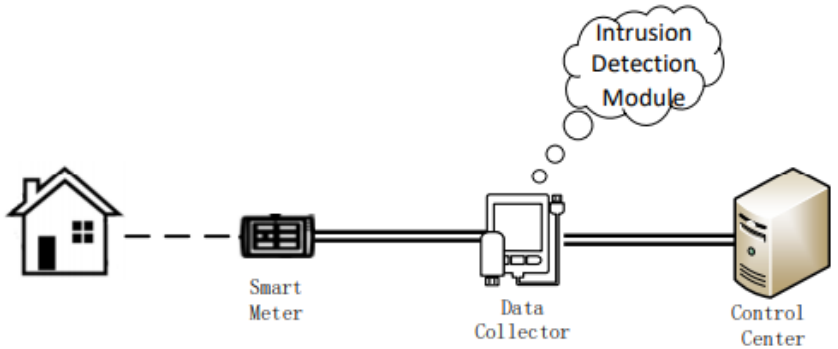


Fig. 1. Smart grid intrusion detection architecture diagram

1. Markov-chain-based game analysis
2. False data injection
3. abnormality
4. Real Time
5. Packet

نقشه راه

۱ مقدمه
- شبکه هوشمند قدرت
- حملات سایبری در شبکه هوشمند
- راهکارهای ارائه شده تاکنون
- راهکار مقاله

۲ آشنایی با داده ها
- بررسی دیتاست
- بررسی یک نمونه

۳ آماده سازی داده ها
- تقسیم نمونه ها به دو کلاس
- حذف نمونه های معیوب
- حذف نمونه های تکراری

۴ پیش پردازش
- استاندارد کردن ویژگیهای پیوسته
- تبدیل ویژگیها با مقدار رشته به معادل عددی

۵ انتخاب ویژگی
- کاهش ابعاد به کمک PCA

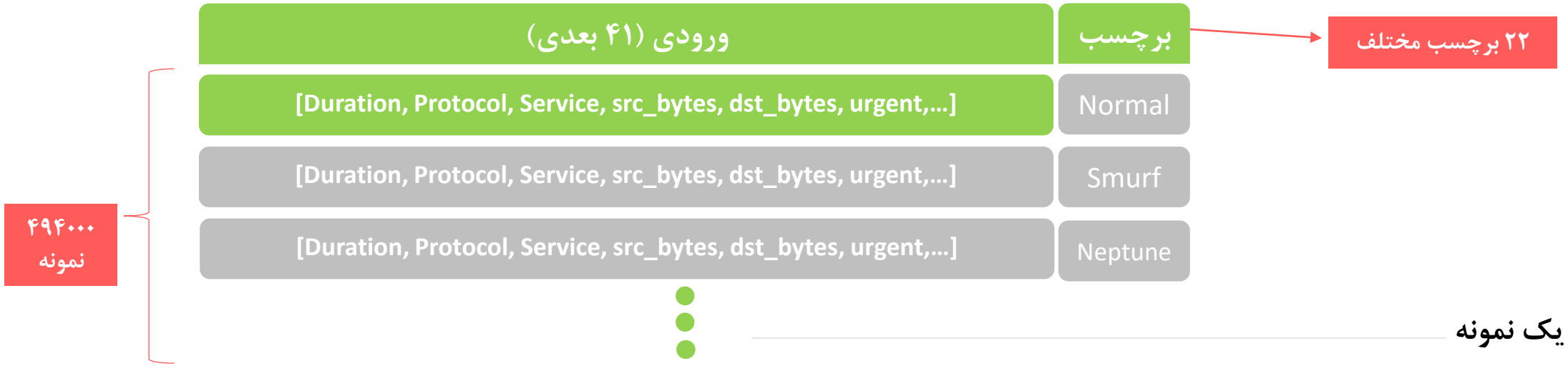
۶ طراحی کلاسیفایر
SVM
Naïve Bayesian
Decision Tree

۷ مقایسه نتایج

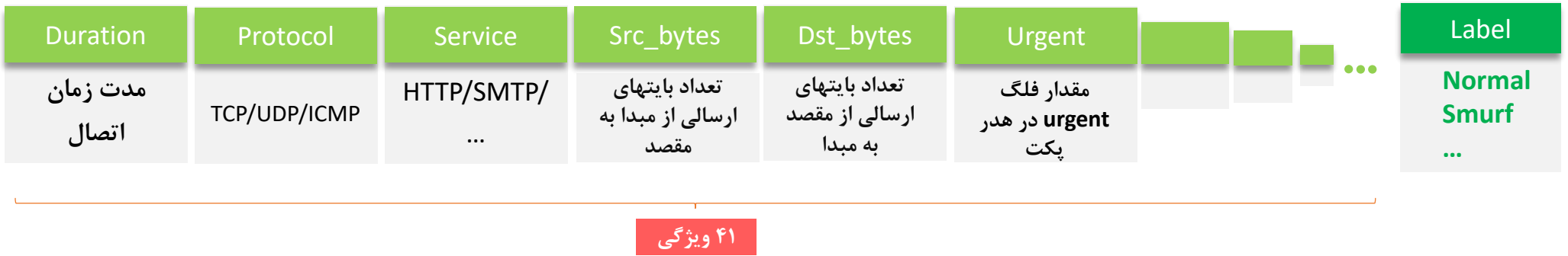
+ پیشنهادها

آشنایی با دیتاست [اطلاعات بیشتر]

- ✓ دیتاست آموزش، شامل حدود ۴۹۴/۰۰۰ نمونه است که یا مربوط به ارتباط عادی و سالم اند و یا به یکی از ۲۲ نوع حمله مختلف تعلق دارند [دریافت]
- ✓ دیتاست تست، شامل حدود ۱۴۰/۰۰۰ نمونه است که علاوه بر حملات موجود در داده های آموزشی،به منظور واقعی تر کردن مدل، شامل نمونه هایی از ۱۴ نوع حمله دیگر نیز میباشد [دریافت]



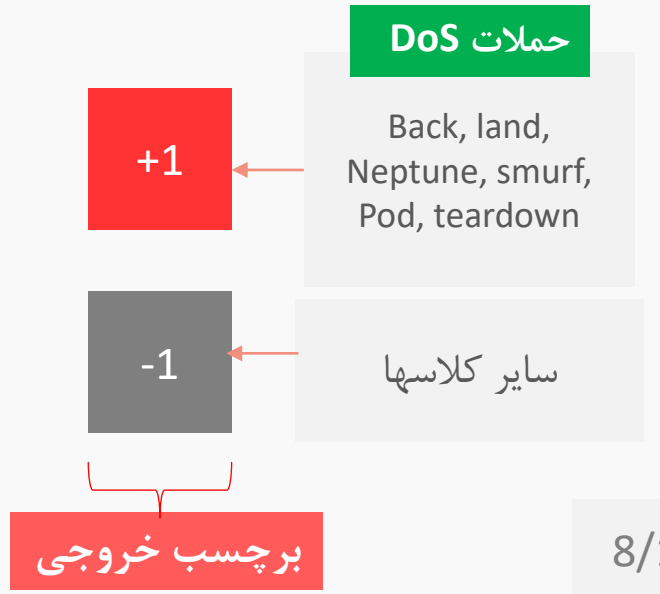
بررسی یک نمونه



آماده سازی داده ها

- ✓ حذف نمونه های معیوب و تکراری
- ✓ ویژگیهایی که مقدار رشته ای دارند به معادل عددی تبدیل میشوند (Sparse Categorical Encoding)

نمونه ها متعلق به ۲۲ کلاس مختلف از حملات اند که تنها ۶ تا از آنها از اقسام حملات DoS هستند لذا نمونه ها باید به دو کلاس تقسیم شوند



پیش پردازش داده ها

تنها پیش پردازش مطرح شده در مقاله، استاندارد سازی ویژگی های پیوسته با توجه به رابطه زیر میباشد

$$X'_{ij} = \frac{(X_{ij} - AVG_i)}{STAND_i}$$

که در ان

AVG_i

میانگین ویژگی i ام

و

$STAND_i$

انحراف از معیاف استاندارد ویژگی i ام است

انتخاب ویژگی و کاهش ابعاد

بنا به گفته مقاله، ۲۹ ویژگی از ۴۱ ویژگی نمونه ها به کمک PCA انتخاب شده اند این ویژگی ها در (TABEL II. FEATURES AFTER FEATURE REDUCTION) در مقاله، لیست شده اند

ویژگهای گسسته انتخاب شده

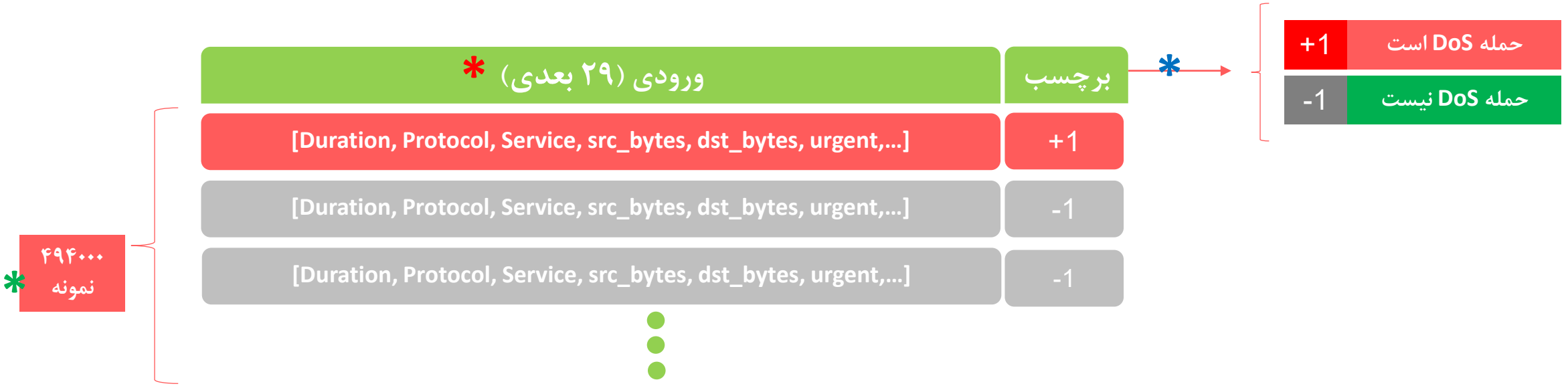
Protocol_type
Service
Flag
Logged_in
Root_shell
Su_attempted
Is_host_login
Is_host_login
Is_guest_login

ویژگهای پیوسته انتخاب شده

Duration, Src_bytesDst_bytes, Urgent, Wrong_fragment, Hot, num_failedLogin_num, Num_root,Num_access_files,num_outbound_cmds,count,srv_count,serror_rate,Srv_serror_rate,rerror_rate,r_error_rate,srv_rerror_rate,Num_file_creation,num_shells,same_srv_rate,diff_srv_rate

* تقریبا نصف نمونه ها تکراری بوده و در مرحله آماده سازی، حذف شده اند
* ۲۹ ویژگی پس از مرحله کاهش ابعاد و انتخاب ویژگی، باقی مانده اند
* نمونه ها در دو کلاس طبقه بندی شده اند

دیتاست پس مراحل گفته شده



طراحی کلاسیفایر

** سرعت پاسخگوی بالا **

هدف: طراحی یک سامانه **بلادرنگ**[1] برای تشخیص حمله است

لذا کلاسیفایر باید

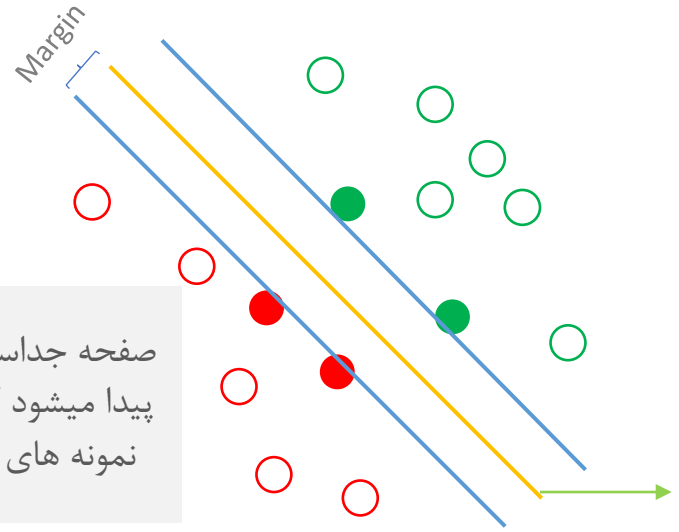
- ✓ ساده باشد تا به راحتی روی بُردهای ارزان قیمت پیاده سازی شود
- ✓ دقت بالایی داشته باشد
- ✓ سرعت پاسخگویی[2] بالایی داشته باشد

مقاله سه کلاسیفایر ساده (NB,SVM,DT)[4] را مقایسه کرده و SVM را به عنوان کلاسیفایر نهایی برگزیده است

1.RealTime System
2.Response Time
3.Supervised Learning

4. NB: Naïve Bayesian
DT: Decision Tree
SVM: Support Vector Machine

$$S(x) = wx + b$$



صفحه جداساز دوکلاس، به گونه ای پیدا میشود که بیشترین فاصله را از نمونه های دو کلاس داشته باشد

- ✓ جز روشهای یادگیری تحت نظارت[3] است
- ✓ منحصرا برای طبقه بندی مسائل دوکلاسه استفاده میشود هرچند میتوان آن را به مسائل چندکلاسی نیز تعمیم داد

چرا SVM ؟

- ✓ ساده است، پس از یافتن پارامترهای آن، طبقه بندی نمونه های جدید صرفا با محاسبه یک ضرب داخلی و تعیین علامت نتیجه آن امکان پذیر است

همین امر SVM را به گیزینه مطلوبی در سامانه های بلادرنگ تبدیل کرده است

معیارهای مقیاسه

پیش بینی شده

واقعی		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

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

- ✓ اگر پکت معیوب، سالم تشخیص داده شود، سیستم در معرض حمله قرار میگیرد
- ✓ اگر پکت سالم، معیوب تشخیص داده شود، پکت لاس خواهیم داشت

مورد اول هزینه بالاتری دارد لذا:

در این مسئله معیار Recall اهمیت بیشتری دارد

نتایج مقاله برای SVM

Classes	Precision	Recall	F1 Score
0	0.97	0.99	0.98
1	0.97	0.97	0.92
Avg/Total	0.97	0.97	0.97

نتایج حاصله برای SVM

Classes	Precision	Recall	F1 Score
0	0.98	0.95	0.97
1	0.98	0.99	0.98
Avg/Total	0.98	0.97	0.97

C. Results Analysis

We use the accuracy, precision, recall and F1 Score indicators to compare three different detection algorithms: Support Vector Machine (SVM), Decision Tree and Naive Bayesian Network. The intrusion detection performance of DoS attacks is shown in Figure 4. As can be seen from the figure, the SVM algorithm achieves the best results.

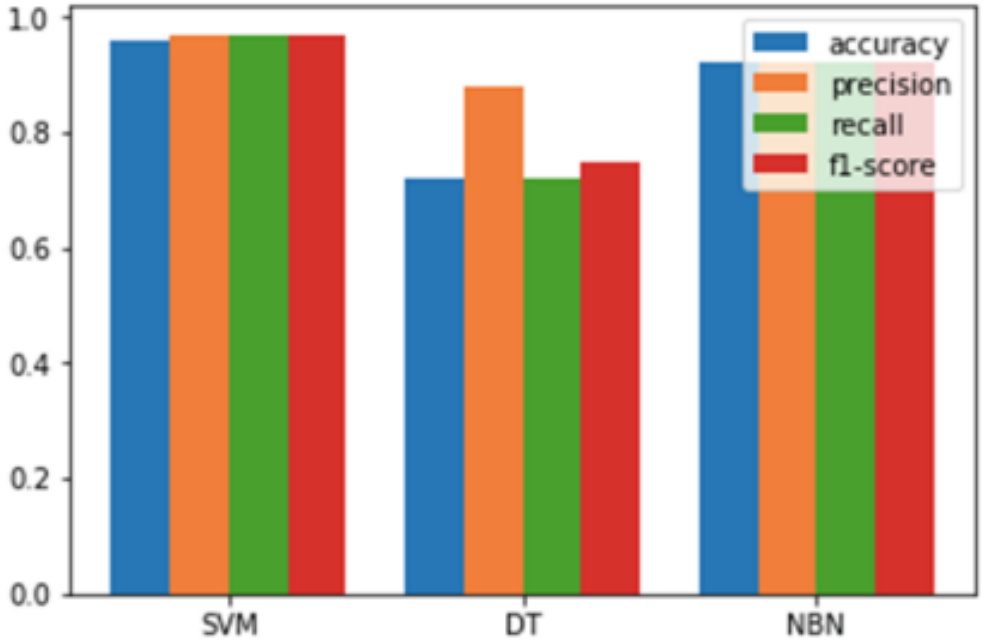


Fig. 4 Results for Different Algorithms

In order to evaluate the classification performance of an algorithm on different categories, we compare the accuracy of each algorithm to the attack and normal categories, so as to recall rate and F1 Score, as shown in TABLE III, TABLE IV and TABLE V. It can be seen that the SVM algorithm has higher precision and recall rate than other algorithms for different detection categories.

نتایج مقاله

TABLE III. SVM DETECTION RESULTS

Classes	Precision	Recall	F1_Score	Support
0	0.97	0.99	0.98	229853
1	0.97	0.87	0.92	60593
Avg/Total	0.97	0.97	0.97	290446

TABLE IV. DECISION TREE DETECTION RESULTS *

Classes	Precision	Recall	F1_Score	Support
0	1.00	0.65	0.79	229853
1	0.43	0.99	0.60	60593
Avg/Total	0.88	0.72	0.75	290446

TABLE V. NAIVE BAYESIAN NETWORK DETECTION RESULTS

Classes	Precision	Recall	F1_Score	Support
0	0.96	0.94	0.95	229853
1	0.80	0.84	0.82	60593
Avg/Total	0.92	0.92	0.92	290446

* احتمالا ۰.۹۷ بوده است

پیشنهادها

- ✓ چنانچه در سایت مربوط به دیتاست گفته شده است، حمله DoS از معدود حملاتی است که الگو مشخصی داشته و قابل تشخیص است.
- ✓ اگرچه امکان طراحی یک کلاسیفایر-چند کلاسی، برای طبقه بندی سایر حملات وجود دارد (حتی با دقت حدود ۰.۹۸ مثلا با شبکه عصبی) اما وجود تنوع و فقدان ساختار و الگو در سایر حملات، مدلهای حاصله را در عمل بی فایده میکند
- ✓ امروزه، جدیدترین روشهای مورد بحث در حوزه امنیت سایبری، عموما برپایه روش های یادگیری تقویتی[1] نظیر POMDP[2] هستند
- ✓ **مدلهای یادگیری تقویتی،پیچیدگی بالایی داشته و معمولا پیاده سازی آنها در قالب یک ماژول سخت افزاری ارزان قیمت امکان پذیر نیست**
- ✓ **ترکیب روشهای یادگیری تقویتی با پارادایم هایی که امروزه در حوزه اینترنت اشیاء[3] و طراحی نرم افزار محبوبیت دارند، نظیر:**

Micro Service Architecture / (Distributed/Cloud/ Edge) Computing

شاید راه حلی برای پیاده سازی مدلهای یادگیری تقویتی،در قالب سامانه های بلادرنگ[4] و با هزینه مناسب، به رغم پیچدگی آنها باشد

1. Reinforcement Learning
2. Partially Observable Markov Decision Process
3. Internet Of Things
4. Real Time