رَبِّ اجْعَلْ هَٰذَا بَلَدًا آمِنًا

پروردگارا، این سرزمین را محلی امن و آسایش قرار ده

بقره ۱۲۶

سُورَةُ القَدْرِ

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

إِنَّآ أَنزَلْنَٰهُ فِى لَيْلَةِ ٱلْقَدْرِ ١ وَمَآ أَدْرَىٰكَ مَا لَيْلَةُ ٱلْقَدْرِ ٢

لَيْلَةُ ٱلْقَدْرِ خَيْرٌ مِّنْ أَلْفِ شَهْرٍ ٣ تَنَزَّلُ ٱلْمَلَٰٓئِكَةُ وَٱلرُّوحُ فِيهَا

بِإِذْنِ رَبِّهِم مِّن كُلِّ أَمْرٍ ٤ سَلَٰمٌ هِىَ حَتَّىٰ مَطْلَعِ ٱلْفَجْرِ ٥

اللهم صل علی محمد و آل محمد و عجل فرجهم

رسول اکرم صلی الله علیه و آله و سلم

هر کس مشتاق بهشت است برای انجام خوبی ها سبقت می گیرد

نهج الفصاحه ص ۹۲ ح ۱۴

عنوان مقاله:

Static Malware Detection in Windows Executables Using Deep Neural Networks and Custom Binary Features

پژوهشگران: سجاد رضایی،دکتر علی فانیان

تاریخ: ۱۴۰۴/۰۷/۱۵

بخش ۱
مقدمه و کلیات

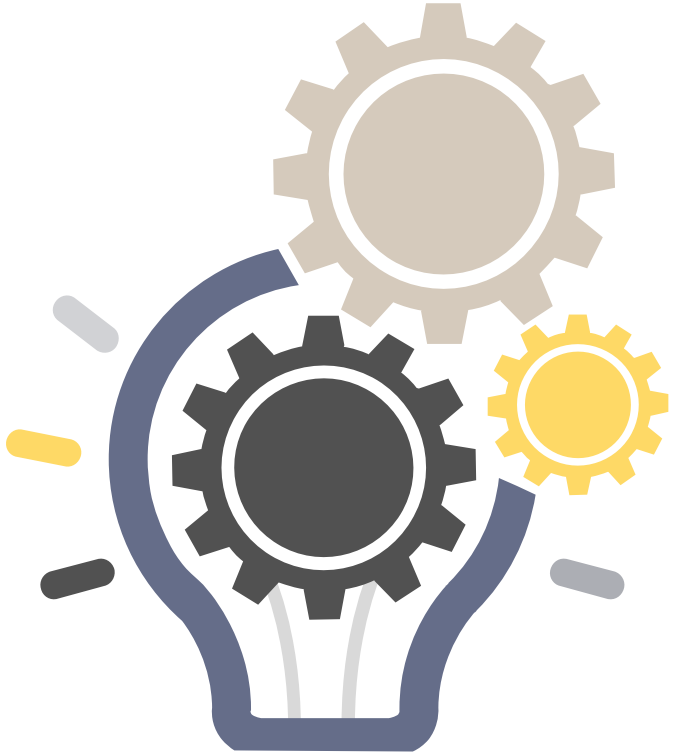
بیان مسئله و هدف پژوهش

۱ نارسایی روش های سنتی شناسایی بدافزارمبتنی بر امضا و تحلیل رفتاری

۲ تحلیل ایستان با منابع پردازشی کمتر و عدم خطر اجرای کد مخرب

۳ شناسایی بدافزارها در فایل های اجرایی با فرمت PE با روش ترکیبی مبتنی بر تحلیل ایستان، یادگیری عمیق و استخراج ویژگی‌های جدید

پیشینه پژوهش



مبتنی بر یادگیری با ویژگی‌های مهندسی‌شده

استفاده از ویژگی‌های استخراج‌شده دستی (مثل سرآیند PE، توابع API و ...) به‌همراه الگوریتم‌های کلاسیک مانند SVM، Random Forest و KNN.

مبتنی بر یادگیری عمیق و استخراج خودکار ویژگی‌ها

استفاده از شبکه‌های عصبی مانند CNN، LSTM و MLP برای تحلیل مستقیم داده‌های‌خام کشف الگوهای پنهان بدون نیاز به مهندسی دستی ویژگی‌ها.

تحلیل ترکیبی و پیشرفته

ترکیب تحلیل ایستان و پویا، به‌کارگیری تکنیک‌هایی نظیر یادگیری تقویتی، شبکه‌های گرافی و ..

بخش ۲
روش‌شناسی

مراحل انجام پژوهش

۱
جمع‌آوری داده

۲
استخراج ویژگی‌ها از فایل‌های اجرایی

۳
پیش‌پردازش داده‌ها

۴
مدل‌سازی و آموزش

۵
ارزیابی عملکرد مدل‌ها

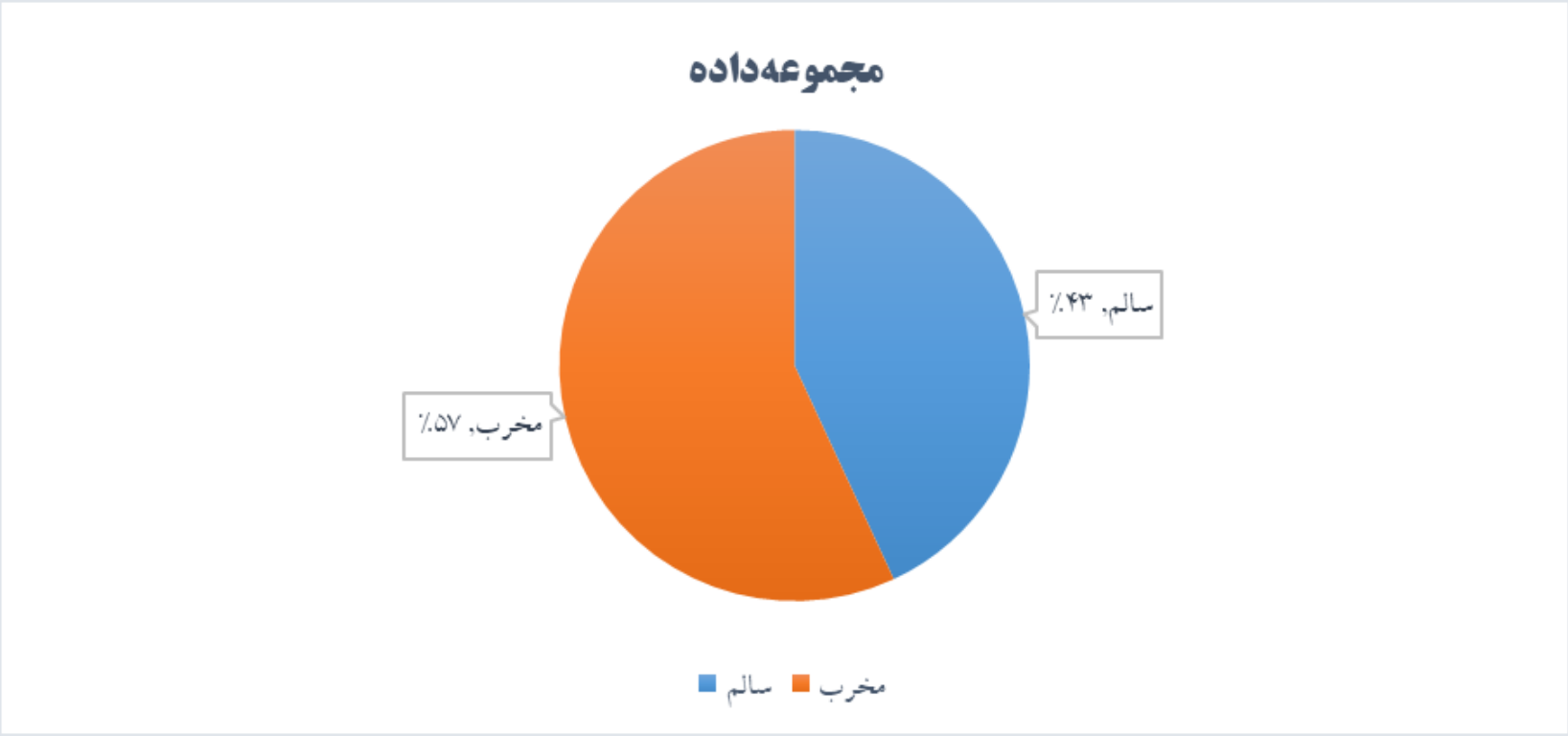
۶
تحلیل نتایج

جمع‌آوری داده و پیش‌پردازش

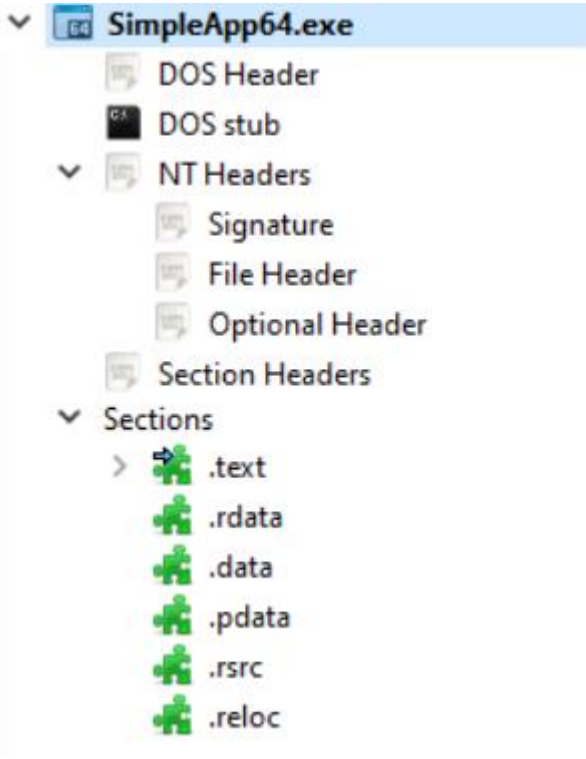
آمار کلی دیتاست

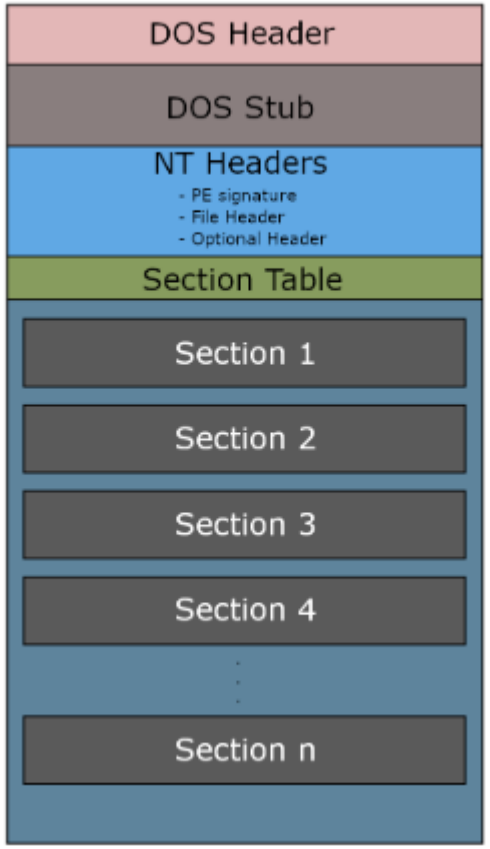
ویژگی	مقدار
تعداد کل فایل‌ها	201,549 فایل
فایل‌های سالم	86,812
فایل‌های مخرب	114,737
حجم فشرده‌شده	43.8 گیگابایت
حجم واقعی	117 گیگابایت
فرمت فایل‌ها	PE (exe, dll, scr)

جمع آوری داده و پیش‌پردازش



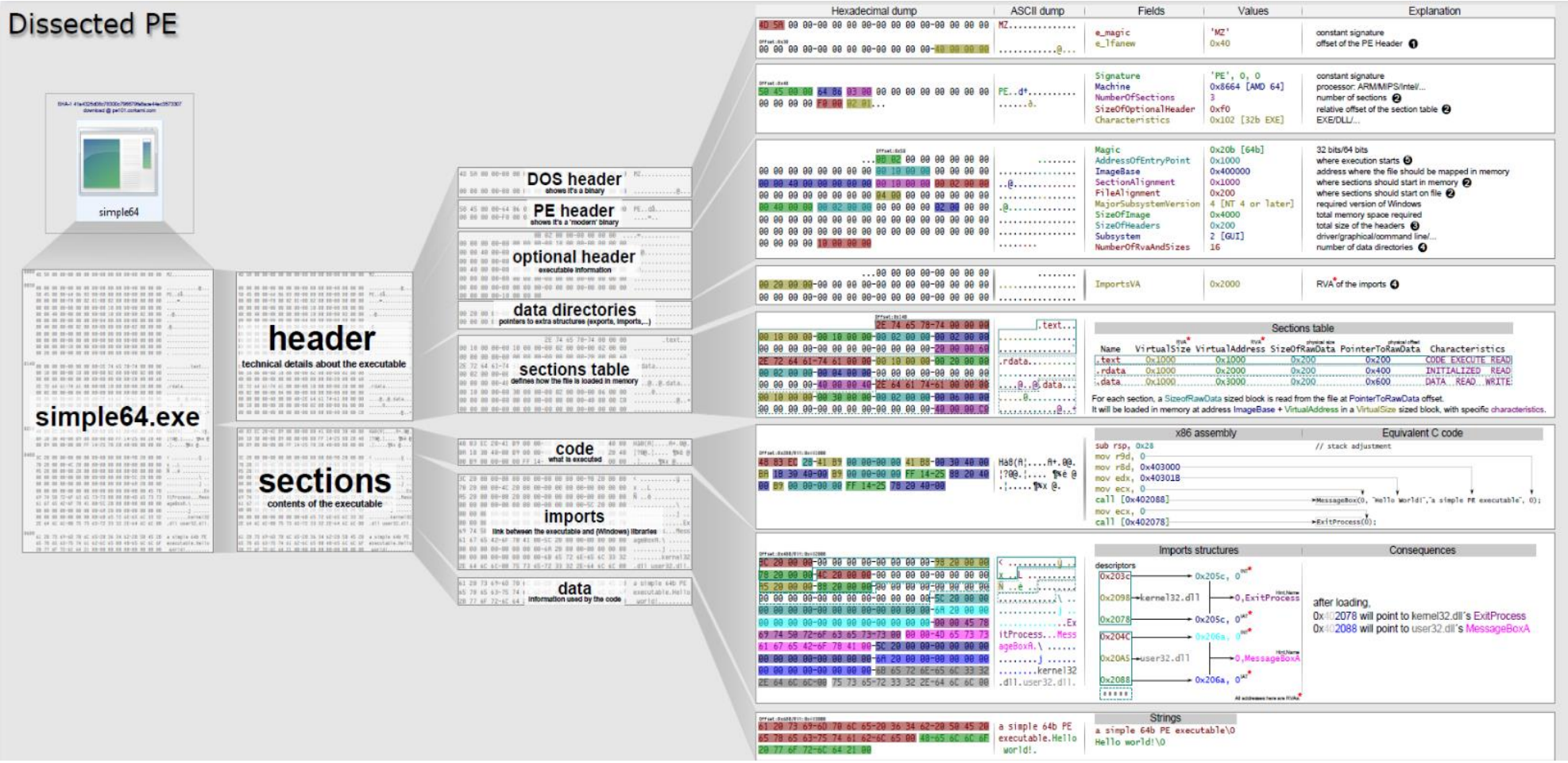
ساختار فایل اجرایی (PE)





باز کردن فایل اجرایی فوق با ابزار pe Bear

ساختار فایل اجرایی2 (PE)



all_feature.html

دسته‌بندی ۷۷ ویژگی استخراج شده از فایل‌های Portable Executable

۷۷ ویژگی کل

۶ دسته اصلی

ویژگی‌های هدر
۱۵ ویژگی

اطلاعات کلی و ساختاری فایل PE

کلیدی: Machine، TimeDateStamp، NumberOfSections، Characteristics، OptionalHeaderMagic، DataDirectoryBits

ویژگی‌های آنتروپی
۹ ویژگی

سنجش تصادفی بودن برای تشخیص پکینگ

کلیدی: FileEntropy، TextEntropy، DataEntropy، SectionsMeanEntropy، ResourcesEntropy

ویژگی‌های بخش‌ها
۱۱ ویژگی

تحلیل sections و نام‌های مشکوک

کلیدی: SectionsNb، SuspiciousSections، NormalSections، SectionsMeanRawsize، HasEmptySectionName

Import/Export
۶ ویژگی

تحلیل وابستگی‌ها و API calls

کلیدی: ImportsNbDLL، ImportsNb، ExportNb، VersionInformationSize

حافظه و منابع
۱۷ ویژگی

مدیریت حافظه و اندازه‌های مختلف

کلیدی: SizeOfCode، SizeOfImage، SizeOfHeaders، ResourcesNb، AddressOfEntryPoint

امنیت و پرچم‌ها
۱۹ ویژگی

پرچم‌های characteristics و تنظیمات امنیتی

کلیدی: ExecutableImage، DLL، LargeAddressAware، HasChecksum، DebugStripped

🚀 نوآوری‌های تحلیل فایل‌های PE

Feature Engineering Innovations in PE File Analysis

۱

ویژگی ترکیبی DataDirectoryBits

بررسی ۶ بخش کلیدی، EXPORT، DEBUG: PE، BASERELOC، RESOURCE، LOAD_CONFIG، TLS و تبدیل به یک عدد

- ✓ بدافزارها DEBUG حذف می‌کنند، فایل‌های سالم LOAD_CONFIG دارند
- ✓ مثال: ۰۱۱۱۱۱ (باینری) → ۳۱ (دسیمال)
- ✓ ۶ ویژگی → ۱ عدد فشرده برای ML

۲

استفاده ترکیبی از Entropy چندمنبع

ترکیب ابزار خارجی ENT با pefile برای محاسبه ۳ نوع entropy مختلف

- ✓ Shannon entropy کل فایل (ent)
- ✓ Entropy سکشن .text
- ✓ Entropy سکشن .data

۳

تفکیک کامل Characteristics

هر فایل PE یک عدد ویژگی دارد که ۱۶ پرچم مختلف در آن پنهان است (مثل "قابل اجرا"، "۳۲ یا ۶۴ بیت"، "دیباگ حذف شده" و...). ما با ماسک بیتی هر پرچم را جداگانه استخراج می‌کنیم

- ✓ IMAGE_FILE_EXECUTABLE_IMAGE: آیا قابل اجرا است؟
- ✓ IMAGE_FILE_DEBUG_STRIPPED: آیا اطلاعات دیباگ حذف شده؟
- ✓ IMAGE_FILE_LARGE_ADDRESS_AWARE: حافظه بالای ۲GB؟
- ✓ ۱۶ ویژگی مستقل به جای یک عدد کلی

۴

نسبت HeaderToCodeSizeRatio

ویژگی جدید برای تشخیص فایل‌های با ساختار غیرعادی و کد پنهان

- ✓ نسبت اندازه هدر به کد
- ✓ تشخیص کد مخفی در هدر
- ✓ شناسایی ساختارهای مشکوک

۵

تحلیل هوشمند اسامی سکشن‌ها

بررسی الگوی نام‌گذاری سکشن‌ها برای تشخیص تکنیک‌های پنهان‌کاری

- ✓ SuspiciousSections: سکشن‌های غیرمعمول
- ✓ HasEmptySectionName: سکشن‌های بی‌نام
- ✓ تشخیص نام‌گذاری جعلی

۶

تفکیک عمیق Resources

استخراج ویژگی‌های پیشرفته از منابع فایل PE فراتر از تعداد ساده

- ✓ ResourcesMeanEntropy
- ✓ ResourcesMaxSize
- ✓ تشخیص منابع رمزگذاری‌شده

🛠 فرآیند پیش‌پردازش و تمیزسازی داده‌ها

Data Preprocessing & Feature Cleaning Workflow

۱ 📊 بارگذاری داده

خواندن فایل CSV و بررسی ابعاد

- تعداد نمونه‌ها
- تعداد ویژگی‌ها
- نوع داده‌ها

۲ 🔍 تحلیل ویژگی‌ها

شناسایی انواع ویژگی‌ها

- ویژگی‌های باینری (۰/۱)
- ویژگی‌های ثابت
- مقادیر −۱ (مفقود)

۳ 🧹 تمیزسازی

پاکسازی و آماده‌سازی

- جایگزینی −۱ با میانگین
- حذف ویژگی‌های غیرمفید
- حذف ویژگی‌های تکراری

۴ 📈 تحلیل همبستگی

ارزیابی اهمیت ویژگی‌ها

- همبستگی با برچسب
- رسم Heatmap
- انتخاب ویژگی‌های مهم

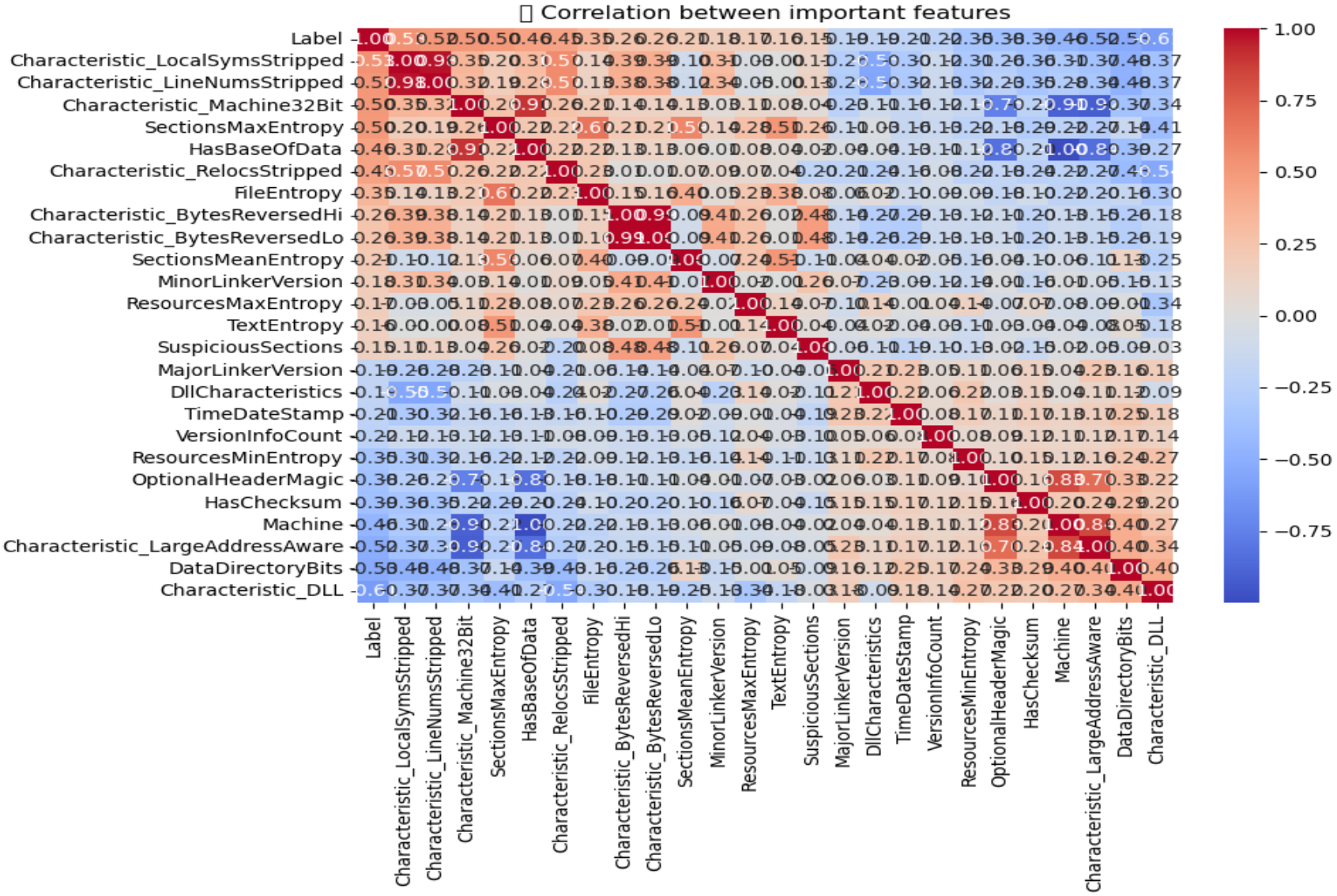
🗑 ویژگی‌های حذف شده

Filename شناسه فایل	ImageBase آدرس تصادفی	Characteristics تجزیه شده
VersionInformationSize همیشه صفر	Characteristic_ExecutableImage همیشه ۱	HasEmptySectionName همیشه ۱
LoaderFlags اکثراً صفر		

✅ نتایج تحلیل

ویژگی‌های باینری شناسایی شدند	ویژگی‌های ثابت حذف شدند	مقادیر −۱ با میانگین جایگزین
همبستگی > ۰.۱۵ ویژگی‌های مهم		

بررسی همبستگی ویژگی ها



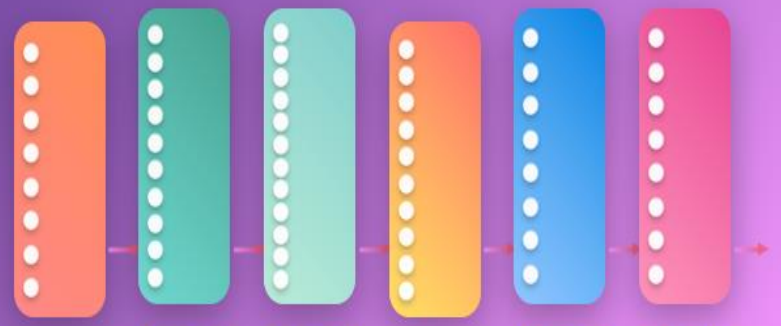
بخش ۳
مدل پیشنهادی

معماری شبکه عصبی عمیق پیشرفته

چندلایه پرسپترون با بهینه‌سازی بیزین

لایه ورودی

ویژگی‌های استاندارد شده
داده‌های نرمال‌سازی شده
تقسیم: ۸۰/۲۰



لایه مخفی ۱
نورون: ۳۸۴
فعال‌ساز: ReLU
حذف: ۰.۱

لایه مخفی ۲
نورون: ۶۴۰
فعال‌ساز: ReLU
✓ :BatchNorm
حذف: ۰.۳

لایه مخفی ۳
نورون: ۷۶۸
فعال‌ساز: GELU
✓ :BatchNorm
حذف: ۰.۳

لایه مخفی ۴
نورون: ۶۴۰
فعال‌ساز: SELU
حذف: ۰.۴

لایه مخفی ۵
نورون: ۳۸۴
فعال‌ساز: SELU
✓ :BatchNorm
حذف: ۰.۴

لایه مخفی ۶
نورون: ۳۸۴
فعال‌ساز: SELU
✓ :BatchNorm
حذف: ۰.۳

لایه خروجی

نورون: ۱
فعال‌ساز Sigmoid
طبقه‌بندی دوتایی

بهینه‌سازی هایپرپارامتر

الگوریتم	بهینه‌سازی بیزین
تعداد آزمایش	۱۰۵
فضای جستجو	چندبعدی
زمان بهینه‌سازی	۹ روز ۱ ساعت

تنظیمات آموزش

بهینه‌ساز	Adam
نرخ یادگیری	۱۰-۴
اندازه دسته	۵۱۲
حداکثر دوره	۱۰۰

تکنیک‌های تنظیم‌سازی

- تنظیم‌سازی L2 ✓
- لایه‌های حذف ✓
- نرمال‌سازی دسته‌ای ✓
- توقف زودهنگام ✓

بهینه‌سازی بیزین - مقایسه پارامترها

از فضای جستجوی اولیه تا پارامترهای بهینه نهایی

🎯 پارامترهای ورودی

فضای جستجوی تعریف شده

🏗 ساختار شبکه

تعداد لایه‌ها	۳ تا ۶
نورون‌ها	۱۲۸ تا ۱۰۲۴

⚡ توابع فعال‌سازی

گزینه‌ها	ReLU, SELU, GELU

🔧 تنظیم‌سازی

Dropout	۰.۱ تا ۰.۵
L2	۱e-۵ تا ۱e-۲
BatchNorm	True/False

⚙ بهینه‌ساز

نوع	Adam, Nadam, RMSprop
نرخ یادگیری	۱e-۴, ۵e-۴, ۱e-۳

بهینه‌سازی بیزین ۱۰۵ آزمایش

🏆 نتایج بهینه

پارامترهای انتخاب شده نهایی

🏗 ساختار نهایی

تعداد لایه	6 لایه
لایه ۱	۳۸۴
لایه ۲	۶۴۰
لایه ۳	۷۶۸
لایه ۴	۶۴۰
لایه ۵	۳۸۴
لایه ۶	۳۸۴

⚡ توابع فعال‌سازی

لایه ۱، ۲	ReLU
لایه ۳	GELU
لایه ۴، ۵، ۶	SELU

🔧 تنظیم‌سازی نهایی

Dropout	۰.۱ تا ۰.۴
BatchNorm	لایه ۲، ۳، ۵، ۶

⚙ بهینه‌ساز نهایی

نوع	Adam
نرخ یادگیری	۱e-۴

بخش ۴
تجزیه و تحلیل

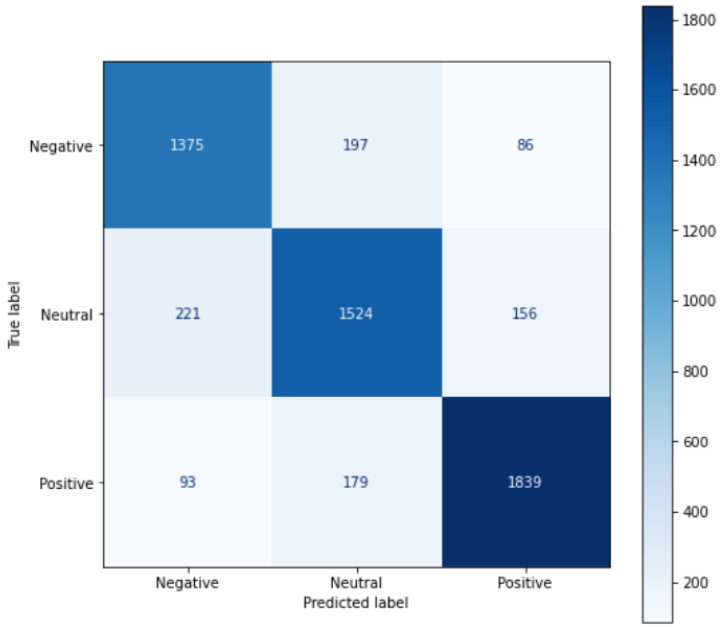
معیارهای ارزیابی

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

$$F1 - score = (2 \times precision \times recall)/(precision \times recall)$$

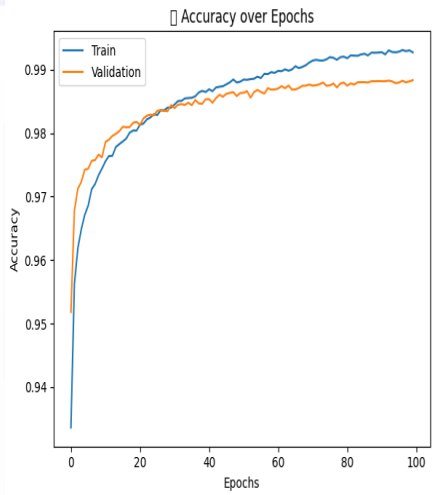
$$Precision = \frac{TP}{TP + FP}$$

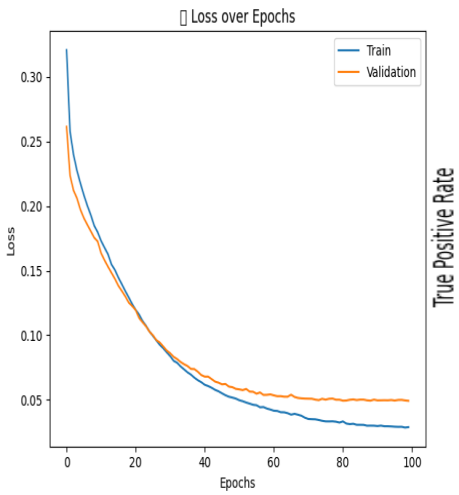
$$Recall = \frac{TP}{TP + FN}$$

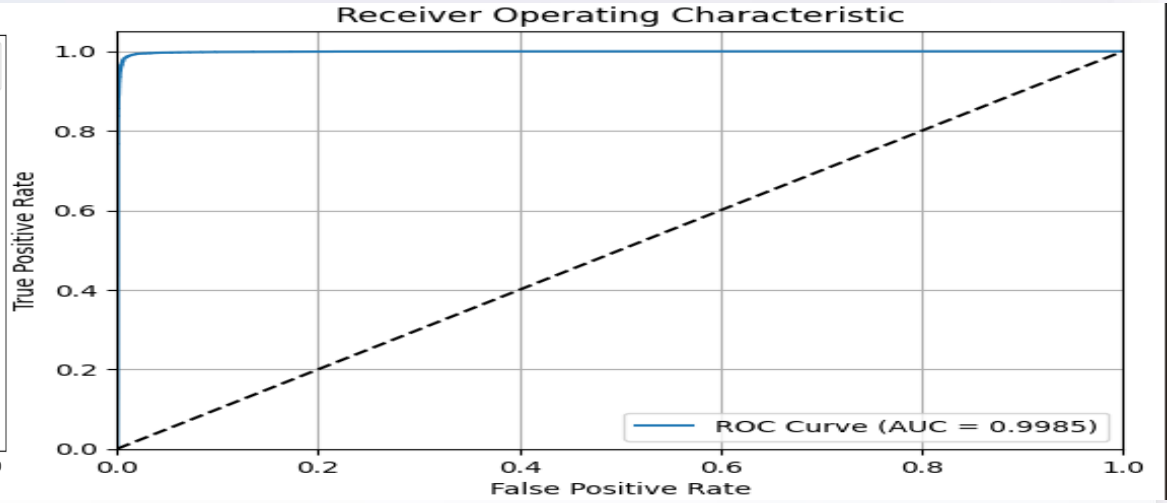


ارزیابی مدل پیشنهادی MLP

نتایج کامل عملکرد شبکه عصبی چندلایه







متریک‌های عملکرد

شاخص‌های کلیدی ارزیابی مدل

۹۸.۸۳% دقت (Accuracy)	۹۹.۰۷% صحت (Precision)
۹۸.۸۸% فراخوان (Recall)	۹۸.۹۸% امتیاز F1 (F1-score)
۰.۰۲۷ تابع زیان	۹۹.۸۵% مساحت زیر منحنی

تمام متریک‌ها بالای ۹۸٪ ← عملکرد بسیار قوی مدل در تشخیص بدافزار

ماتریس درهم‌ریختگی

Confusion Matrix

	پیش‌بینی ۰ (سالم)	پیش‌بینی ۱ (بدافزار)
واقعی ۰ (سالم)	۱۷۱۴۰ True Negative	۲۱۲ False Positive
واقعی ۱ (بدافزار)	۲۵۸ False Negative	۲۲۶۹۳ True Positive

سیستم تشخیص بدافزار

آپلود فایل‌ها برای تشخیص بدافزار

فایل‌های PE (اجرایی ویندوز) خود را آپلود کنید تا توسط 4 مدل مختلف تحلیل شوند.

فایل‌های خود را اینجا بکشید یا کلیک کنید

پسوندهای پشتیبانی شده: .exe, .dll, .sys

انتخاب فایل‌ها

انتخاب مدل‌ها:

- RANDOM_FOREST — مدل یادگیری ماشین
- LIGHTGBM — مدل یادگیری ماشین
- XGBOOST — مدل یادگیری ماشین
- MLP — مدل یادگیری ماشین

انتخاب همه | لغو انتخاب همه

شروع تحلیل

Search
ENG
12:12 PM
9/11/2025

منابع و مراجع مهم

1. Alkhateeb, E. G., A.; Habibi Lashkari, A. (2024). Identifying Malware Packers through Multilayer Feature Engineering in Static Analysis. Information https://doi.org/10.3390/info15020102

2. Ammar Ahmed E. Elhadi, M. A. M., Ahmed Hamza Osman. (2012). Malware Detection Based on Hybrid Signature

3. Aula Hamed, N. A.-o. (2025). Advanced Framework for Detecting Malware in Portable Executable (PE) Files Using a Multi-Model.

4. Behaviour Application Programming Interface Call Graph. American Journal of Applied Sciences. https://doi.org/10.3844/ajassp.2012.283.288

5. Bengio, Y., Goodfellow, I., & Courville, A. (2017). Deep learning (Vol. 1). MIT press Cambridge, MA, USA.

با سپاس فراوان از

حضار محترم