Malware Detection with Random Forest & MongoDB

Team-2 [CSE-A]

D Darshana - CSE20020
K Arun Kumar - CSE20032
M Sree Chandana - CSE20042

INDEX

Introduction to Malware Detection

Definition

Malware is a major threat to computer systems and networks which can cause significant damage to data and software, as well as compromise the security of an organization.
Thus, malware detection is essential for protecting against these threats.

Algorithm

Random Forest is a machine learning algorithm that has been shown to be effective in detecting malware.

Database

MongoDB is flexible schema and scalable which make it well-suited for handling the variety and volume of data often encountered in ML applications.

Role of MongoDB

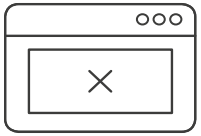
MongoDB is a NoSQL database that is commonly used in big data applications. It is ideal for storing and processing large amounts of data quickly and efficiently. In Malware detection, MongoDB can be used to store and analyze large volumes of data about malware samples and their behavior.

By using MongoDB in conjunction with Random Forest, Malware detection systems can achieve high levels of accuracy and speed. The combination of these technologies allows for real-time detection and response to new and emerging threats.

Existing works

Title	Algorithm	Year	Accuracy
Android Malware detection	Naïve Bayes	2017	80%
Windows malware detection.	SVM, Decision Tree, Random Forest	2018	93%
Android Malware using deep neural networks	Deep neural networks	2019	97.3%
Android malware using traditional ML algorithms	Combination of traditional machine learning algorithms	2020	96.7%

Proposed Model



Dataset

Training / Testing

Output

Storage in MongoDB

A. ATTRIBUTES OF DATASET:

	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp	e_csum	...	SectionMaxChar
0	0.0	-0.038591	-0.050297	-0.041557	-0.040212	-0.042419	0.148298	-0.016139	-0.036843	-0.031918	...	1.076024
1	0.0	-0.038591	-0.050297	-0.041557	-0.040212	-0.042419	0.148298	-0.016139	-0.036843	-0.031918	...	0.097299
2	0.0	-0.038591	-0.050297	-0.041557	-0.040212	-0.042419	0.148298	-0.016139	-0.036843	-0.031918	...	0.097299
3	0.0	-0.038591	-0.050297	-0.041557	-0.040212	-0.042419	0.148298	-0.016139	-0.036843	-0.031918	...	0.097299
4	0.0	-0.038591	-0.050297	-0.041557	-0.040212	-0.042419	0.148298	-0.016139	-0.036843	-0.031918	...	0.097299

5 rows × 77 columns

e_magic: "Magic number", 2-byte value that identifies the file as a PE file.

e_cblp: "Bytes on last page", number of bytes in the last block of the file that are not used.

e_cp: "Pages in file", number of blocks in the file (part of EXE file).

e_crlc: "Relocations", the number of relocation entries in the file.

e_sp: "Stack pointer", the initial value of the SP register.

e_csum: "Checksum", a 16-bit checksum of the entire file.

e_ip: "Instruction pointer", the initial value of the IP register.

e_cs: "Code segment", the initial value of the CS register.

e_lfarlc: "File address of relocation table", the file offset of the relocation table.

e_lfanew: "File address of new exe header", the file offset of the start of the PE header.

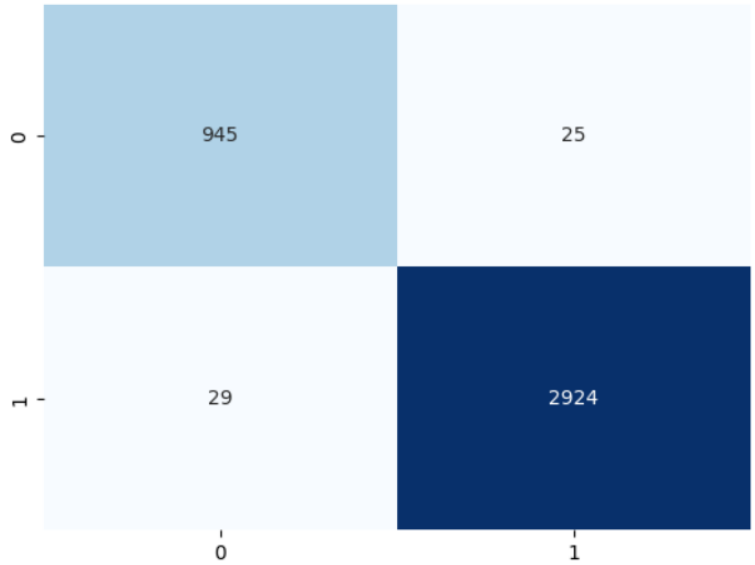
B. TRAINING & TESTING

Once the dataset has been collected, it can be used to train the model using Random Forest model. The model will learn to identify patterns and behaviors that are indicative of malware. Once trained, the model can be used to detect new malware samples with a high degree of accuracy.

Train Rate of model = 0.2 [80% training & 20% testing]

CLASSIFICATION REPORT:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	970
1	0.99	0.99	0.99	2953
accuracy			0.99	3923
macro avg	0.98	0.98	0.98	3923
weighted avg	0.99	0.99	0.99	3923



C. OUTPUT

	Name	0	1
0	Skype-8.10.0.9.exe	0.989924	0.010076
1	vlc-3.0.2-win64.exe	0.771497	0.228503
2	stinger32.exe	0.180000	0.820000
3	SpotifyFullSetup.exe	0.721422	0.278578
4	uftp_english.exe	0.480280	0.519720
5	161a59f2525518f799c63f916c80fe85f50c5b09c74dc2...	0.170380	0.829620
6	eaa478e65696ad5cbdb42c1b4bd6954f2a876fdde2e519...	0.002452	0.997548
7	reverse_shell.exe	0.020258	0.979742
8	873b9eaef6ea5ed6126086594529a3395bdbc5d63c97d8...	0.051043	0.948957
9	ScratchInstaller1.4.exe	0.352534	0.647466
10	69eb27dd3bbf5077dcd795872535b89af9a898254b90ad...	0.026440	0.973560
11	3334686141a400bb522824fa6f7faf30614372fe11837a...	0.001992	0.998008
12	3ec4cb928846f8298e5a13b3e96bfc2a709cb3b005a31e...	0.002447	0.997553
13	252f705dc15d7a305afd3e0619fa014c10b679248f71b7...	0.023168	0.976832
14	wordweb8.exe	0.202659	0.797341
15	c89f1e55b418a4447394994498971c6e6f3848bfe39ef9...	0.051043	0.948957
16	winrar-x64-550.exe	0.615234	0.384766

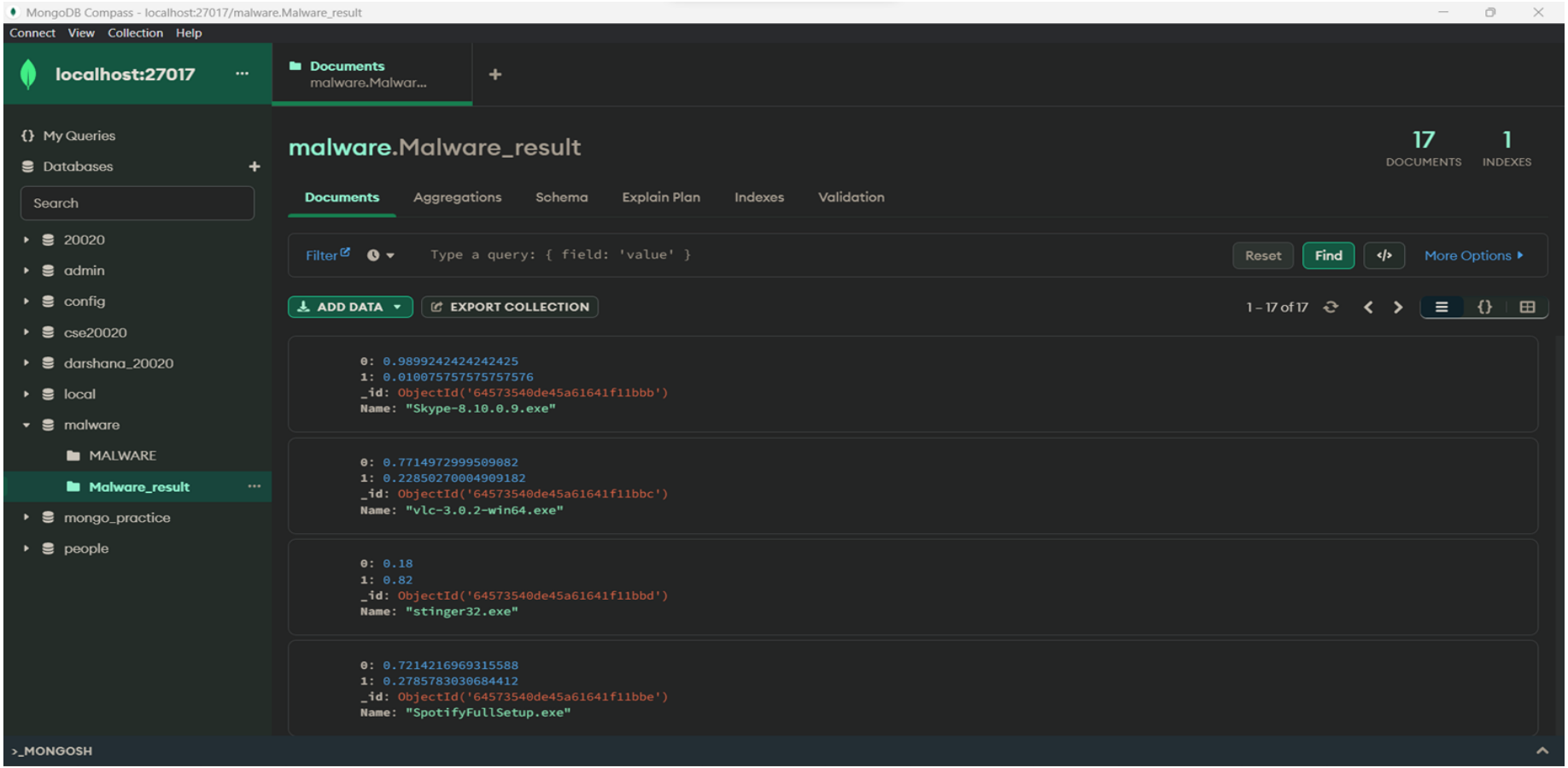
ACCURACY

```
from sklearn.metrics import accuracy_score

acc = accuracy_score(y_pred, y_test)
print("Accuracy:", acc)
```

```
Accuracy: 0.9936273260260005
```

D. FINAL OUTPUT STORED IN MONGODB



Challenges in Malware Detection

Despite the effectiveness of Random Forest and MongoDB in Malware detection, there are still challenges that must be addressed.

1. Ability of malware to evolve and adapt to new detection methods.

2. **To overcome this challenge**:
Malware detection systems must be constantly updated and refined. This requires ongoing research and development, as well as collaboration between security researchers and industry professionals.

Conclusion: The Future of Anti Malware

Malware detection is an essential component of modern cybersecurity. By leveraging the power of machine learning algorithms like Random Forest and databases like MongoDB, we can develop highly accurate and efficient systems for detecting and responding to malware threats.

As technology continues to evolve, the threats we face. Nevertheless, by staying ahead of the curve and investing in research and development, we can ensure that our anti malware detection systems remain effective and reliable for years to come.

THANK YOU