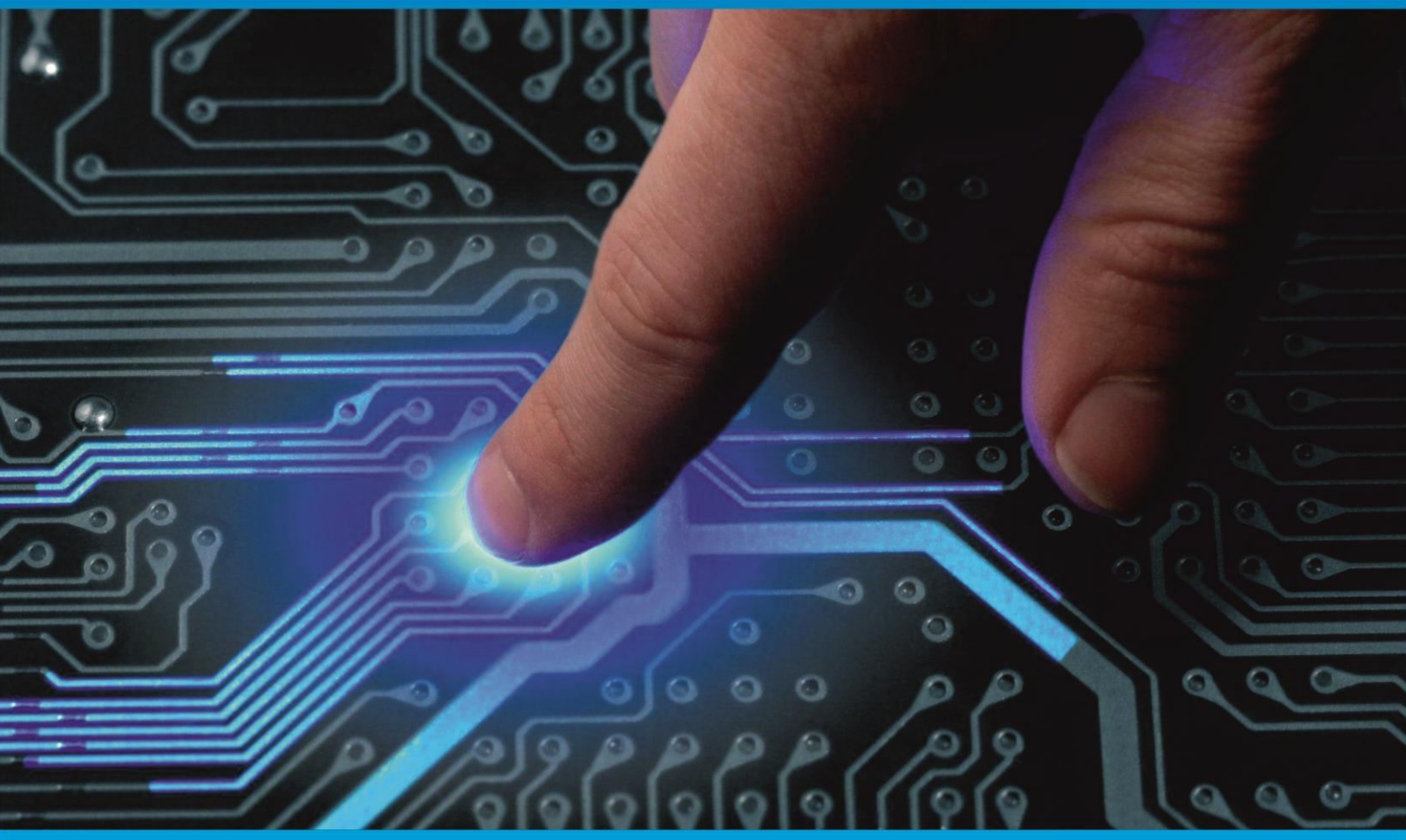
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 5, May 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

Malware Classification Using Deep Learning

Shikku Thomas, Ms.Mrinmoyee,

P.G. Student, Department of Computer Science, St. Joseph's College Bangalore, India

Assistant Professor, Department of Computer Science, St. Joseph's College Bangalore, India

ABSTRACT: Malware, short for Malicious Software, is increasing in both number and sophistication as our digital footprint expands. It is a serious issue to which many efforts are being made. In today's cyber security landscape, malware detection is crucial. There are numerous machines. In order to detect malware automatically, machine learning algorithms are used. Recently, there has been an issue. Deep Learning has recently been employed with improved performance In the real world, deep learning models perform much better. Long sequences of system calls are analyzed. A superficial analysis is used in this study. The feature extraction method (word2vec) is based on deep learning. in order to depict any malware based on its opcodes. And, For the classification challenge, the Gradient Boosting technique for malware classification was employed. Validation is done using K-fold cross-validation. Without sacrificing a validation split, model performance can be improved. Observation With only a small sample size, the results suggest a 96 percent success rate.

KEYWORDS: Machine learning, deep learning, supervised learning, classification, malware detection

I.INTRODUCTION

Malware is malicious software that causes harm to a computer system or device when it is run. The repercussions of executing malware can range from small to devastating, including the following:a nuclear power plant failure or the loss of key data As a result Malware has a constant impact on our daily lives and computer.Systems must be protected from malware attacks 24 hours a day, seven days a week.

Automatic malware detection is a scientific research subject, with a variety of machine learning methods being utilised and developed to solve the problem. Deep learning algorithms, which are multilayer neural networks, have recently been utilised in the machine learning field, and they have proven to be successful in a variety of learning processes, including classification. Deep Learning, on the other hand, takes longer to train and retrain models, which is prevalent in the malware detection industry because new malware types develop regularly and must be added to the training sets. The trade-off is thus difficult: traditional machine learning algorithms are fast but not very accurate, whereas developing deep learning methods are time demanding but extremely accurate. Malware detection is more precise.The signature method is an anti-virus approach that has been used for a long time. for the detection of malware A signature is a brief string of bytes that identifies a person. Identifies a malware type in a unique way. This strategy, however, is ineffective. because of the ever-changing forms of malware. Malware analysis is divided into two stages. During the initial phase, the so-called Malware is first detected and identified during the discovery phase. In The second phase, malware classification, is where security comes into play .Each danger sample is identified or classified as one of the threats by the systems. malware families that are appropriate Feature is used to classify malware. In the classification phase, vector selection approaches are applied. There are two forms of analysis: static and dynamic analysis.The dynamic analysis approach entails running the malware and observing its actions, as well as noting changes in the executed environment. The complexity of environment setup is very high and the time needed to see the outcome of executing all malware is too long. However, this approach gives the most safe, good, and reliable results. The static analysis process, on the other hand, entails looking at the metadata of malware executables, assembly code instructions, and binary data in the malware's material. This method requires a simple environment setup cost and the results can be obtained much faster. The accuracy of malware classification is strongly reliant on the use of appropriate feature vectors. Malware classification will be done using deep neural network architectures in this study. A Convolutional Neural Network-based feature extraction of malware, an autoencoder-based feature learning, or a recurrent neural network-based classifier can all be used to accomplish deep learning-based malware classification. There are multiple features representing malware in each of the aforementioned works, and each of these is merged for higher classification accuracy. For malware representation, we employ a shallow deep learning network based on the Word2Vec vector space model, and for malware classification, we employ a gradient search technique based on the Gradient Boosting Machine

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

II.BACKGROUND

There are numerous papers in the literature that use data mining and machine learning techniques to detect malware automatically .Heuristic static malware detection approaches were employed in SAVE and SAFE , and they are considered some of the early studies in this area, encouraging malware detection experts. These studies suggested that patterns may be used to detect harmful material in executable files. Other approaches for detecting patterns from the header of portable executables (PE), the body of PE, or both are later developed. Pattern detection can also be done on bytecode or by disassembling the code, extracting opcodes, and looking for patterns in the opcodes for malware.Drew et al. employed Strand, a gene sequence classifier that offers a robust classification technique for unstructured material of any alphabet, including source code and compiled code.To show Stand's suitability for categorising, we used machine code.Viruses. Texts are employed in the classification of malware. Word2vec develops text word embeddings in the form of word vectors, which are then used to classify texts using these vector representations. After feeding machine instructions through the Capstone disassembler and generating opcodes, Popov utilized word2vec to construct word embeddings from malware opcodes. Then, using a convolutional neural network-based classifier, these word vectors are utilized to classify executable programmes. With up to 97 percent success, the system is tested with a small dataset of 2400 portable executables (PE) of only two classes (malicious vs. bening). Using word2vec for PE representation, we used a similar approach to malware classification.

III.PROPOSED METHOD

3.1. DATASET

During the 2015 WWW Conference, Microsoft disclosed a big malware dataset.
Microsoft Malware Classification Challenge Dataset(BIG 2015)1 is the name of the dataset.This dataset was utilised in this study. The training dataset contains 10869 samples of malware from nine different malware types.
Description related to different malware families is discussed below:

- **Ramnit:** It steals user credentials such as password credit card information and halts security software.
- **Lollipop:** Lollipop is an adware that displays advertisements in the browser and allows a hacker to monitor user traffic.
- **Kelihos_ver1 and Kelihos_ver3:** Trojan types can fully control user pc and spread by sending spam email from user pc to others.
- **Vundo:** It could be responsible for pop-up ads and installing other malicious content.
- **Simda:** These type of malware snatch the passwords from user pc and create a backdoor for hackers.
- **Traceur:** Author earns revenue by displaying fake advice on search engines using this malware attack.
- **Obfuscator.ACY:** These are considered as obfuscated malware, and their purpose could be any of the malware mentioned above.
- **Gatak:** This is also a type of Trojan that seems legitimate but infects a computer with its malicious code.

3.2. CLASSIFICATION ARCHITECTURE

The malware attributes that will be utilised to represent each malware sample for the classification task have a direct impact on the accuracy and execution time of the classification task. In this case, employing a malware classification architecture section, our malware classification architecture Will be based on the Word2Vec vector space representation.be clarified.

3.3. MALWARE REPRESENTATION

Malware samples can be represented in a variety of ways, and some of them will be discussed in this section. When reading malware code, hex codes and assembly notation can be used. The order of events The accumulation of successive 16-bytes in hexadecimal digits is known as hexadecimal digits.words similar to those in Listing 1.
Listing 1: Assembly code / Opcode

```
0 x401180  55 31 C0 89 E5 B9 11 00
           00 00 57  56 8D 55 A4 53
```

The first value provides the starting address, while the subsequent numbers are memory machine codes, and each value (byte) is an element for the PE, such as an instruction. data or codes The Interactive Disassembler (IDA) [6] is a disassembler that allows you to interact with your code. reverse engineers binary apps and analyses them automatically

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

awareness of cross-references between code parts of the API call stack, and other data In IDA, for example, a byte As illustrated in Listing 2, the sequence can be examined.

Listing 2: Assembly view

```
0 x401180 : 5 5                 push ebp
0 x401181 : 3 1 c0              xor eax , eax
0 x401183 : 8 9 e5              mov ebp , esp
0 x401185 : b9 11 00 00 00 mov ecx , 0 x11
0 x40118a : 5 7                 push e d i
0 x40118b : 5 6                 push e s i
0 x4 0 1 1 8 c : 8 d 55 a4      l e a edx , dword
0 x 4 0 1 1 8 f : 5 3            push ebx
```

3.4. WORD2VEC

The disassembled codes, known as opcodes , are assembly codes that represent binary codes in a human-readable format. To extract the assembly codes, we detach them from their arguments. Only the sequences of assembling operations are included. The final version of the the example presented in Listing 3 is an example of an opcode sentence.

Listing 3: Concatenated OpcodeWords

```
push  xor  mov  mov  push  push  l e a  push
```

word2vec [7], which has lately acquired prominence in the analysis of natural language text, is the proposed feature representation methodology employed in this research. Word2Vec is a programme that creates embeddings for words extracted from natural language text At the conclusion of this course, you will be able to:Word embeddings (vector representations of words) are created throughout this procedure. For each input instance, a unique code is generated. These vectors depict the syntactic structure as well as semantic links between words; words that have similar meanings In the vector space, common context is close to each other.A number of parameters are used for building Word2Vec representations that are listed in Table 1.

vector length	The length of the word vectors.
window length	Length of the context window word.
sample sent frequency	The frequency of words. Higher frequency words will be down sampled
learning initial frequency	Initial learning frequency.
training iterations	The number of training epochs.

3.5. CLASSIFICATION

Several classification systems have been proposed by the scientific community over the years. Previous expertise in various tasks often guides the selection of the most suited classifier for a given assignment.domains, as well as by trial-and-error methods. Nonetheless,Recently, a group of academics assessed the performance of roughly 180 people.Gradient Boosting Machine [4] (for Regression and Classification) is a forward learning ensemble method, on the other hand. The guiding principle is that with increasingly finer approximations, good forecasting outcomes can be produced. GBM creates regression trees progressively on all of the dataset's characteristics in a fully distributed manner, that is, each tree is generated in parallel.For the classification task, a number of parameters are employed to form a GBM.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

3.6. EVALUATION MEASURES

Specifically, precision and logarithmic loss. The percentage of correct predictions is measured by accuracy. In most cases, accuracy alone isn't enough to determine the predictability of a prediction; we additionally look at the predictability of the prediction.The logarithmic loss (logloss) is a sensitive indicator of The concept of probabilistic confidence is incorporated into the accuracy. It's the cross entropy between the true label distribution and the true label distribution.as well as the expected probability It is, as indicated in Equation 1, the The model's negative log likelihood.

$$\text{Log loss} = -1/N \sum_{i=1}^{N} \sum_{j=1}^{M} Y_{ij} \log (P_{ij})$$

where N equals the number of observations, M equals the number of class names, log equals the natural logarithm, and yi j equals 1 if observation I belongs to class j.

IV.EXPERIMENT & RESULT

In this work, we experimented with 4 separate sampled datasets.There are 8 different malware classes in experiment dataset. These are listed in Table 2.

Table 2: Experiment Setup Parameters

Experiment No	Sample per Class	Total Samples
1	100	800
2	200	1600
3	300	2400
4	398	3184

To eliminate bias, k-fold cross validation is utilised. The k-value can be set anywhere between 5 and 10. If a higher number is specified, the cross validation time will also be longer. In order to come up with a baseline, you'll need to do some math.The k-fold cross validation parameter was set to 5 in our model for cross validation.The values for the Word2Vec parameter are as follows.200 dpi vector epochs:10, transmitted sample rate:0, starting learning rate:1Experiments are run in parallel on a cluster of three servers, each with a 32-core Xeon processor and 32GB of memory.The logloss value is used to evaluate models. Following the GBM models When a set of models is developed, the one with the least number of parameters is chosen.value of logloss The test dataset findings have confusion matrices.Tables 3, 4, 5, and 6 illustrate the results of eac h of the four studies. Rates of error are also listed in the table's rightmost column. The percentages of errors are at most 6%, indicating that they are clearly successful. As the sample size grows, the error rate decreases (see Figure 2).That's a 50% decrease (roughly) from 0.0688 to 0.0386. As a result, larger malware sample data will improve the results even further.

Table 3: Experiment 1 - Cross Validation (5 fold) ConfusionMatrix

Class	1	2	3	4	5	6	7	8	Error	Rate
1	89	5	0	1	0	0	3	2	0.1100	11/100
2	7	90	0	0	2	0	1	0	0.1000	10/100
3	0	0	97	2	0	0	0	1	0.0300	3/100
4	1	1	0	97	1	0	0	00	0.0300	3/100
5	2	0	0	3	93	0	0	2	0.0700	7/100
6	1	0	0	0	0	95	1	3	0.0500	5/100
7	4	1	0	1	2	1	90	1	0.1000	10/100
8	1	0	1	3	1	0	0	94	0.0600	6/100
Total	105	97	98	107	99	96	95	103	0.0688	55/800

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

Table 4: Experiment 2 - Cross Validation (5 fold) ConfusionMatrix

Class	1	2	3	4	5	6	7	8	Error	Rate
1	177	7	0	2	4	1	5	4	0.1150	23/200
2	6	186	1	1	3	0	0	2	0.0653	13/199
3	0	0	198	2	0	0	0	0	0.0100	2/200
4	0	3	0	196	0	0	0	1	0.0200	4/200
5	6	2	0	0	191	0	0	1	0.0450	9/200
6	0	0	0	2	0	198	0	0	0.0100	2/200
7	8	2	1	6	1	0	181	1	0.0950	19/200
8	1	1	0	2	1	0	0	0195	0.0250	5/200
Total	198	201	200	211	200	199	186	204	0.00482	77/1599

Table 5: Experiment 3 - Cross Validation (5 fold) ConfusionMatrix

Class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Error**	**Rate**
1	265	12	0	1	2	0	13	6	0.1137	34/299
2	10	284	0	3	1	0	0	0	0.0470	14/298
3	0	0	298	1	0	0	0	1	0.0067	2/300
4	0	2	0	298	0	0	0	0	0.0067	2/300
5	5	0	0	0	287	0	0	4	0.0401	12/299
6	2	0	0	6	0	292	0	0	0.0267	8/300
7	20	2	0	6	2	1	265	3	0.1137	34/299
8	4	2	0	0	1	0	3	290	0.0333	10/300
Total	306	302	298	318	293	293	281	304	**0.0484**	116/2395

Table 6: Experiment 4 - Cross Validation (5 fold) ConfusionMatrix

Class	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Error**	**Rate**
1	365	12	0	1	8	0	8	4	0.0829	33/398
2	15	381	0	0	0	0	1	1	0.0427	17/398
3	0	0	397	1	0	0	0	0	0.0025	1/398
4	0	2	0	396	0	0	0	0	0.0050	2/398
5	4	0	0	1	388	1	0	4	0.0251	10/398
6	1	1	0	7	1	387	1	0	0.0276	11/398
7	28	3	0	8	1	1	354	3	0.1106	44/398
8	1	0	0	1	2	0	1	393	0.0126	5/398
Total	414	399	397	415	400	389	365	405	**0.0386**	123/3184

V. CONCLUSION

We employed sequences of opcodes without arguments to provide a new malware representation method based on static analysis in this paper. We've demonstrated that by using low-dimensional Word2Vec feature vectors and a boosting classifier like GBM, malware classification accuracy may easily reach 94% to 96%. The accuracy may be higher if the cross-validation k-fold value is bigger. It should also be noted that the GBM model tree depth can be searched using grid search (tree pruning), which would allow for a wider range of model lookups in the GBM tree search, potentially improving accuracy. Hyperparameter optimization steps will be included to the model search in the upcoming, and the dataset will be expanded to cover every type of malware found in the wild. Discovering semantic links among malware opcodes will be a study path.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 5, May 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0905123 |

REFERENCES

[1]Mihai Christodorescu and Somesh Jha. 2003. Static Analysis of Executables to Detect Malicious Patterns. In *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12 (SSYM'03)*. USENIX Association, Berkeley, CA,USA, 12–12. http://dl.acm.org/citation.cfm?id=1251353.1251365

[2]George E Dahl, Jack W Stokes, Li Deng, and Dong Yu. 2013. Large-scale malware classification using random projections and neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 3422–3426.

[3]Jake Drew, Tyler Moore, and Michael Hahsler. 2016. Polymorphic malware de- tection using sequence classification methods. In *Security and Privacy Workshops (SPW), 2016 IEEE*. IEEE, 81–87.

[4]Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine, In Annals of Statistics. *Annals of Statistics* 29, 1189–1232. http:// citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.9093

[5]Wenyi Huang and Jack W Stokes. 2016. MtNet: a multi-task neural network for dynamic malware classification. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 399–418.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details