JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DEEP LEARNING IN MALWARE TYPES

Prachi Gadhire,Dr. Farhat Jummani
Research scholar, Assistant Professor
Computer department,
[1]JJT university, Rajasthan ,India

ABSTRACT:

Malware, or harmful software as it is sometimes called, is always changing to keep up with the expansion of our internet presence. Malware detection receives a lot of attention in the present cybersecurity environment since it is such a serious threat. Numerous machine learning techniques have been used in automatic malware detection during the past year. Recently, deep learning has been used with better results. Deep learning models perform noticeably better when analysing lengthy series of system calls. This work uses the shallow deep learning-based feature extraction method word2vec to represent any given malware based on its opcodes. Also chosen for the classification task is the Gradient Boosting malware classification method. K-fold cross-validation is used to validate the model performance without skipping a validation

Keywords: Malware, classification.

1. Introduction

When malware is executed on a computer system or device, it causes damage. Malware is malicious software. Running malware could have negative effects ranging from trivial to catastrophic, such as losing important data or causing a nuclear power plant to fail. Because malware constantly interferes with our daily lives, our computer systems require constant malware security. Many researchers and cybersecurity companies are developing new, effective techniques and tools for the automatic recognition and detection of malicious software in order to give protection. Many machine learning techniques are utilised and customised for the

automatic malware detection challenge because it is a scientific study topic. Deep learning techniques, which are multilayer neural networks, have recently gained popularity in the machine learning field and are effective in a variety of learning tasks, including classification. Deep Learning, on the other hand, necessitates more computation time to train and retrain the models, which is typical in the malware detection industry given the constant emergence of new malware kinds that must be added to the training sets. The trade-off is consequently difficult: traditional ML machine learning algorithms are quick but not very accurate, whereas new deep learning techniques take longer but are more accurate in detecting malware. An old method of malware detection employed by antivirus software is the signature method. A signature is a brief string of bytes that specifically identifies a particular type of virus. However, because virus varieties are always evolving, this strategy is not very effective. Two stages make up malware analysis. Malware is first discovered and identified in the first phase, which is referred to as the malware discovery phase. Security systems attempt to identify or classify each threat sample as a member of the proper malware family during the second step, which is known as the malware classification phase. Static and dynamic analysis are the two types of feature vector selection approaches that are utilised in the classification process for malwareThe dynamic analysis method is running the virus, watching how it behaves, and noting changes to the environment it is running in. The setup of the environment is really complex, and it takes too long to get the results of running the malware. However, this method produces the most secure, admirable, and trustworthy outcomes. On the other hand, the static analysis method entails studying the malware by scrutinising the metadata of the executables, the assembly instructions, and the binary data within its content. The results can be acquired using this way considerably more quickly, and it just costs a little to build the setting.

RELATED WORK

The use of data mining and machine learning techniques for automatic malware detection is widely studied in the literature. Heuristic static malware detection techniques have been employed in SAVE and SAFE, which are regarded as some of the early efforts in this field and have influenced malware detection researchers. These studies suggested the use of patterns to find harmful code in executable files. Later, various techniques are created to find patterns in the portable executables' (PE) header, content, or both. By disassembling the code, extracting the opcodes, and looking for patterns in the opcodes for malware traces, pattern detection may also be done on the bytecode. Malicious software code is commonly concealed via packing and obfuscation. Recent research has solved this issue by automatically decrypting malware that has been encrypted without making any

prior assumptions about the encryption methods. Drew et al. employed Strand, a gene sequence classifier, to show Stand's suitability for classifying malware. Strand offers a robust classification technique that can easily accommodate unstructured material with any alphabet, including source code or generated machine code. They used the same dataset as us and achieved accuracy scores of 95% and higher with shorter training periods. Santos et al. employed data mining techniques to find malware using the opcodes from executables. Popov suggested using word2vec, a recently created well-liked tool to examine natural language documents, for the classification of malware. Word vectors created by Word2vec are then used to categorise texts by creating word embeddings for each word in the text. Popov passed machine instructions through the Capstone disassembler, which generated the opcodes, and then used word2vec to create word embeddings from the malware opcodes. Then, a classifier based on convolutional neural networks is used to categorise executable codes using these word vectors. A modest dataset of 2400 portable executables (PE) from just two classes—malicious and benign—is used to evaluate the system, with up to 97 percent success.

DEEP LEARNING-BASED MALWARE CLASSIFICATION Neural networks are widely used in research for malware analysis and detection. Dahl et al., for instance, used random projections and neural networks on a broad scale to categorise malware. They have demonstrated, meanwhile, that adding more hidden layers to the neural network does not significantly improve accuracy. Saxe and Berlin classified static malware using feedforward-based deep neural networks. However, the results of dynamic analysis are absent from their study. Obfuscated binary disassembly might not produce sufficient classification inputs. On the other hand, Huang et al. assessed multi-task learning theories and used up to four hidden layers of feedforward neural networks. Recurrent networks are used by Pascanu et al. to represent system call sequencing in order to create a "corpus" for malware opcodes. They test Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) and report remarkably high categorization accuracies. Deep neural network architectures will be used in this effort to classify malware. Convolutional neural networks, autoencoders, and recurrent neural networks are all examples of deep learning-based malware classifiers that can be used to extract features from malware. For higher classification accuracy, each of the aforementioned works has multiple features that each indicate a malware type. For modelling malware, we utilise a Word2Vec vector space model-based shallow deep learning network, and for malware classification, we employ a gradient search technique based on Gradient Boosting Machine.

DATASET

In 2015, Microsoft made available a sizable malware dataset. Microsoft Malware Classification Challenge Dataset is what it's named. This dataset was utilised in this study. 10869 samples of malware from 9 different malware classes are included in the training dataset. The following malware classes are listed in order: Ramnit, Lollipop, Kelihos ver3, Vundo, Simda, Tracur, Kelihos ver1, and Obfuscator. ACY, Gatak (9) In order to demonstrate the effectiveness of created classification models trained with this small numbers of samples, we sampled 100, 200, 300, and 398 files from all classes (apart from class 5) and used these samples. Class 5 (Simda) samples were removed since there were only 42 of them, and we wanted an equal number of samples from each class in the studies to exclude the bias.

CLASSIFICATION ARCHITECTURE

The accuracy and execution time of the classification task are strongly impacted by the choice of malware attributes that will be utilised to represent each malware sample. Malware Representation Various methods can be used to represent malware samples, and some of them will be discussed in this section. When reading malware code, hex codes and assembly notation can be used. The accumulation of successive 16-byte words like the one in Listing 1 results in the hexadecimal digit sequence. Assembly code/Opcode 0 x401180 55 31 C0 89 E5 B9 11 00 00 00 57 56 8D 55 A4 53 is listed in Listing 1. Each value (byte) is a component for the PE, such as data or instruction codes. The starting address is represented as the first value, and the succeeding numbers are the machine codes in the memory. Using cross-references between code parts, knowledge of the API call stack, and other data, The Interactive Disassembler (IDA) [6] does reverse engineering and automatic analysis of binary applications.

EVALUATION MEASURES Accuracy and logarithmic loss are two metrics used to measure classification performance. The percentage of accurate predictions is used to gauge accuracy. Since accuracy alone is frequently insufficient to evaluate how robust a prediction is, we additionally measure the logarithmic loss (logloss), a sensitive measure of accuracy that also takes the idea of probabilistic confidence into account. The cross entropy between the distribution of the true labels and the expected probabilities is what determines the probability. Equation 1 demonstrates that it is the model's negative log likelihood. lодloss is defined as 1 N X N i=1 X M j=1 yij log(pij) (1) where yij equals 1 if observation I belongs to class j, N is the number of observations, M is the number of class labels, log is the natural logarithm,and 0 otherwise, and pij is the predicted probability that observation i is in class j.

CONCLUSION

In this paper, we presented a new malware representation method based on static analysis, we used sequences of opcodes without arguments.

REFERENCE

1 Glisson, W. B., and T. Storer, "Investigating Information Security Risks of Mobile Device Use within Organizations ", Americas Conference on Information Systems (AMCIS), 2013

2. Kamarudin, I. E., S. a. M. Sharif, and T. Herawan, "On Analysis and Effectiveness of Signature Based in Detecting Metamorphic Virus", International Journal of Security and Its Applications, 7(4), 2013, pp. 375-384.

3 Mcmillan, J., W. B. Glisson, and M. Bromby, "Investigating the Increase in Mobile Phone Evidence in Criminal Activities", Hawaii International Conference on System Sciences (HICSS-46), 2013