COMPARISON OF MACHINE LEARNING TECHNIQUES FOR MALWARE DETECTION AND CLASSIFICATION

SUPERCOMPUTER EDUCATION AND RESEARCH CENTER

INDIAN INSTITUTE OF SCIENCE

CONTENT

- Objective
- State of Art
- Dataset
- Methodology
- Experimental results
- Conclusion

OBJECTIVE

The objectives of this work is to:

- *detect zero day malware and classify malware variants in a precise and scalable manner that correlates the static features from executables (PE-32 files) of a known malware family.*

- *identify the appropriate n-gram size and the machine learning algorithm that is best suited for this approach.*

STATE OF ART

- *Detection of malicious code by applying machine learning classifiers on static features: A state-of-the- art survey [Shabtai et al., 2009][1]*
  - A survey that presents numerous approaches available to extract features based on a static attributes of executables for malware detection

- *Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection [Firdausi et al., 2010][2]*
  - Behaviour-based malware detection using five classifiers: k-Nearest Neighbours, Naive Bayes, J48, Decision Trees, SVM and Multilayer perceptron
  - Concludes J48 as the classifier with the best accuracy

- *A Feature Selection and Evaluation Scheme for Computer Virus Detection [Henchiri et al., 2006] [3]*
  - Evaluates the predictive power of a classifier by taking into account dependence relationships that exist between viruses
  - Employs Decision trees, Naive Bayes and SVM to show heuristic features are better than traditional features.

- *Malware Detection using Machine learning and Deep learning [Rathore et al., 2018] [4]*
  - Opcode frequency is the feature vector for both supervised and unsupervised malware classification
  - Random Forest outperforms Deep Neural Network with opcode frequency as a feature.

- *A hybrid deep learning image-based analysis for effective malware detection [Sitalakshmi Venkatraman et al., 2010] [5]*
  - Describes a hybrid model for image-based analysis using eight different distance measures and deep learning architectures.
  - Identifies malware family by adopting images of the distance scores.

- *A multi-level deep learning system for malware detection [Zhong et al., 2019] [6]*
  - Focuses on learning a specific data distribution for a particular group of malware using tree structures
  - Provides a higher accuracy with lower execution time.

- *Malware detection based on deep learning algorithm [Yuxin et al., 2019][7]*
  - Deep belief networks outperforms other machine learning techniques in terms of accuracy in case of unlabeled data
  - Autoencoders :
    - model the underlying structure of input data
    - significantly reduce the dimensions of feature vectors

DATASET

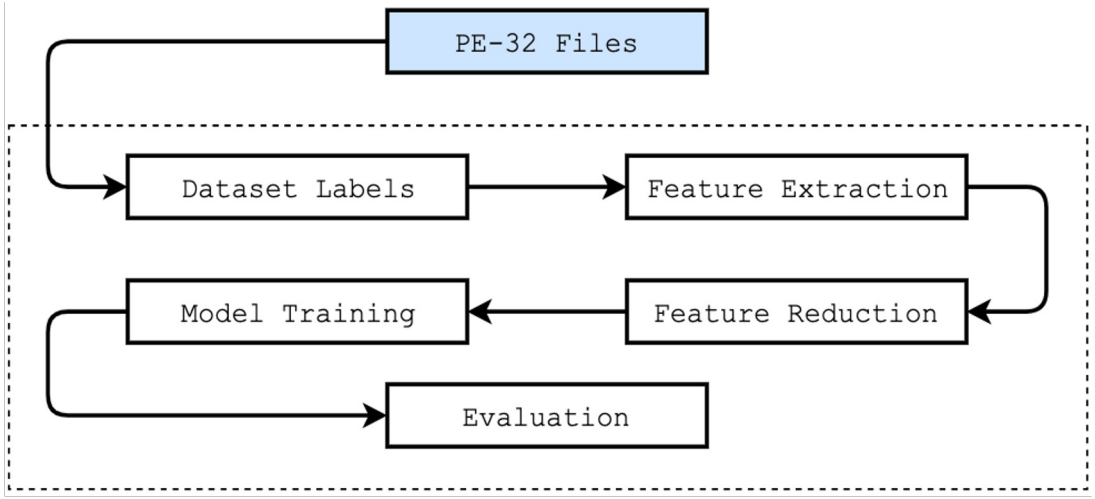
Our dataset consists of a total of 1.08 million Portable Executable (PE32) files. The corpus of PE-32 executables is subjected to a virus scan test using the ClamAV antivirus engine. This splits the entire data into 0.75 million malware binaries and 0.33 million benign files. The dataset has one benign class and the following malware classes:

Trojan Virut	Worm Allaple
Trojan Agent	Win Downloader
Trojan Generic	Trojan Udr
Win Spyware	Trojan Delf
Dropper Agent	Trojan Luder

For malware classification we further divided our dataset to form 3 balanced, even distributed datasets (each consisting of 1100 files, 11000 files and 165,000 files).

METHODOLOGY

The various steps involved in the proposed approach towards malware classification are represented in the form of a block diagram



A. Feature Extraction

- Bloom filter is used to generate the signature for each PE-32 file by extracting unique attributes that are exhibited by each and every malware family member.
- We extract only the code segment from the given executable.
- It generates a bit vector of size 8192 bits, irrespective of the size of the input file.
- The presence of 1's in the Bloom filter output signifies the presence of certain pattern within the code segment.

METHOD

- A stream from the code segment is used to generate a set of n-grams (n-grams of size 4 byte, 8 byte, 16 byte and 32 byte)
- Every n-gram is then fed as an input to the Bloom filter hash function which in turn returns a numeric value
- Depending on this numeric value, the corresponding bit in the Bloom filter is then set to '1'.
- At the end of this step, we get a corpus of bit vectors for every PE-32 executable for each n-gram size.

B. Feature Reduction

Features extracted are reduced by selecting only those bit positions whose value is ‘1’ in the Bloom filter.

C. Model training

The Bloom filter output is fed as a feature vector to the following machine learning classification algorithms:

J48
Random Forest
JRIP
SVM
SMO
Naive Bayes
Multinomial Naive Bayes
Multilayer perceptron
Semi-supervised Generative Adversarial Network

Training phase - The bit vectors corresponding to a malware family are used to train the classifiers model to identify the signature pattern of that malware family.

D. Evaluation Measures

Based on the learning, the model is now employed to classify new unseen malware variants as belonging to a specific known malware family. The output from each of the classification algorithms used was studied for their effectiveness in malware classification. To evaluate our classification model the following evaluation methods are considered:

Precision (P): Precision is defined as the ratio between the number of malware classified correct with respect to the total number of malware classified. Precision is defined as follows:

$$Precision(P) = TP / (TP + FP)$$

Recall (R): Recall is defined as the ratio between the number of malware classified correct with respect to the total number of malware. Recall is computed as follows:

$$Recall(R) = TP / (TP + FN)$$

F-Score: The F-Score is a measure of the accuracy of the classification. It considers both the precision and recall in a single representative score for evaluation purposes. It is defined as:

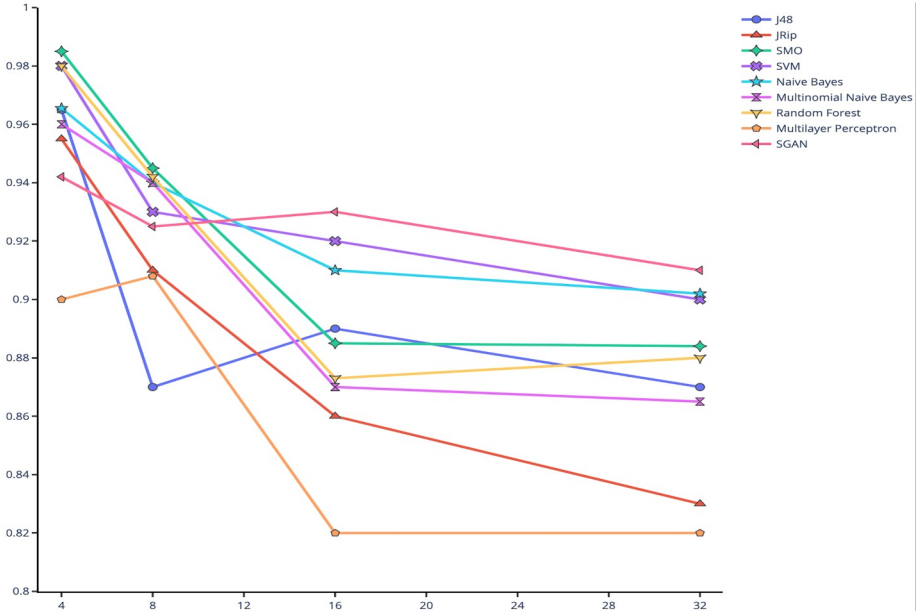
$$F\text{-}Score = 2TP / (TP + FN)$$

EXPERIMENTAL RESULTS

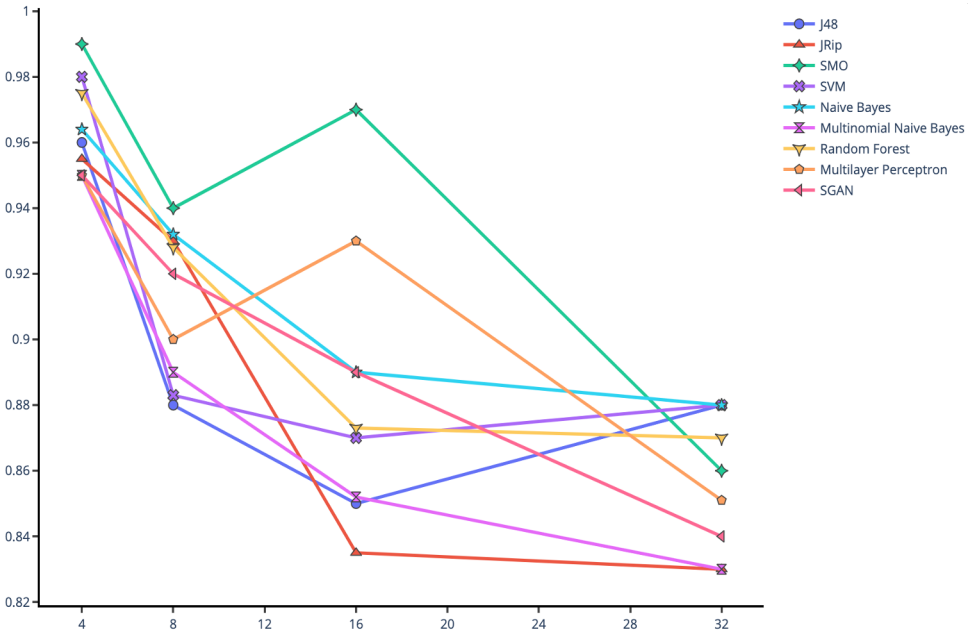
Experiment 1

For the first experiment, we chose a dataset of 1100 files comprising of 100 files each from 10 malware families and one class of benign files.

Precision

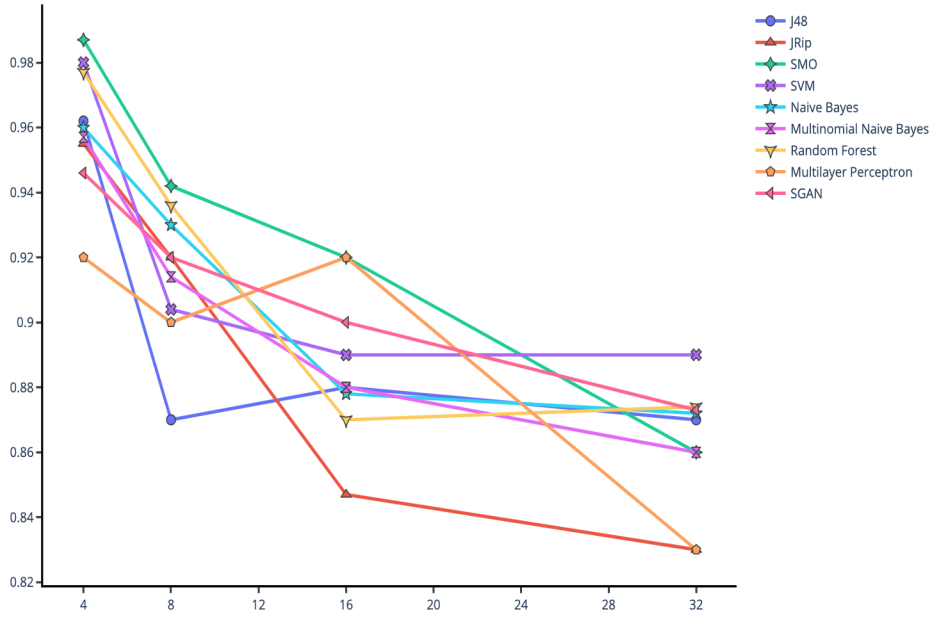


n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.965	0.955	0.985	0.98	0.965	0.96	0.98	0.9	0.942
8	0.87	0.9	0.945	0.93	0.94	0.94	0.942	0.9	0.925
6	0.89	0.86	0.89	0.92	0.9	0.87	0.87	0.82	0.93
32	0.87	0.83	0.87	0.9	0.9	0.865	0.88	0.82	0.9



Recall

n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.96	0.955	0.99	0.98	0.96	0.95	0.975	0.95	0.95
8	0.88	0.93	0.94	0.88	0.93	0.89	0.93	0.9	0.92
6	0.85	0.835	0.97	0.87	0.89	0.85	0.87	0.93	0.89
32	0.88	0.83	0.86	0.88	0.88	0.83	0.87	0.85	0.84



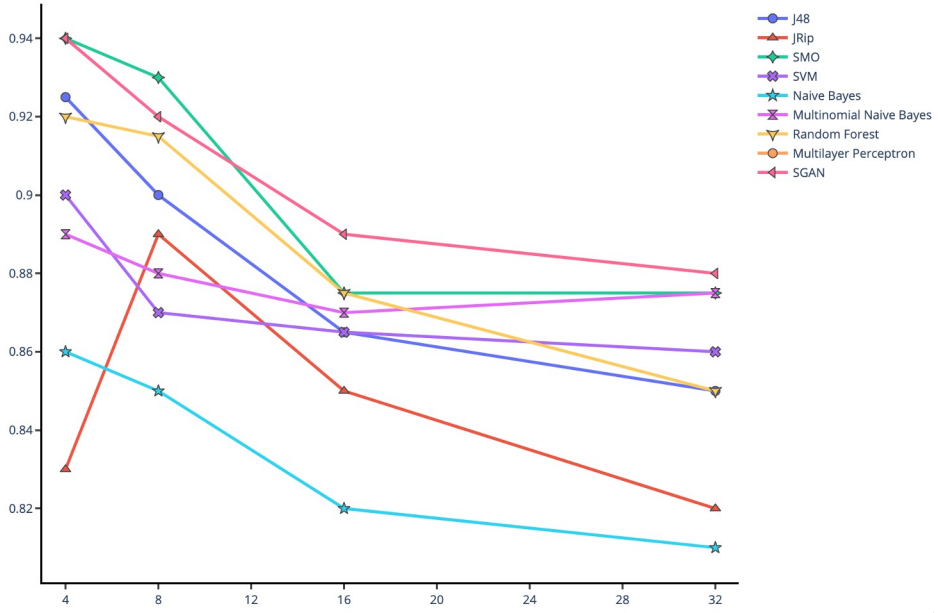
F-Score

n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MUTLINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.962	0.955	0.987	0.98	0.96	0.957	0.977	0.92	0.946
8	0.87	0.92	0.942	0.904	0.93	0.94	0.936	0.9	0.92
6	0.87	0.847	0.92	0.89	0.88	0.88	0.87	0.92	0.9
32	0.87	0.83	0.86	0.89	0.87	0.86	0.87	0.83	0.87

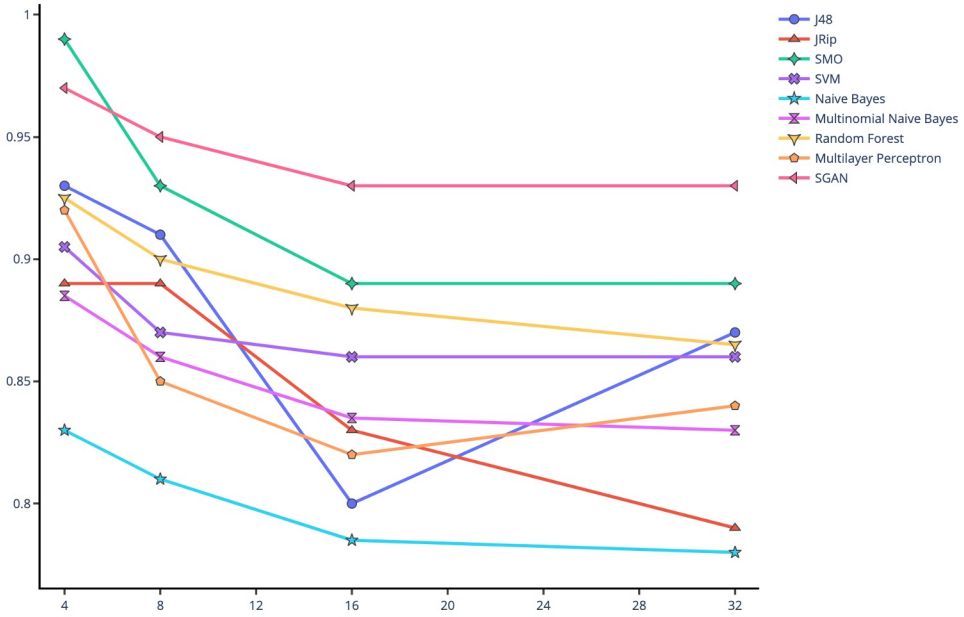
Experiment 2:

For this experiment, we chose a dataset of 11000 files comprising of 1000 files each from 10 malware families and a class of benign file.

Precision

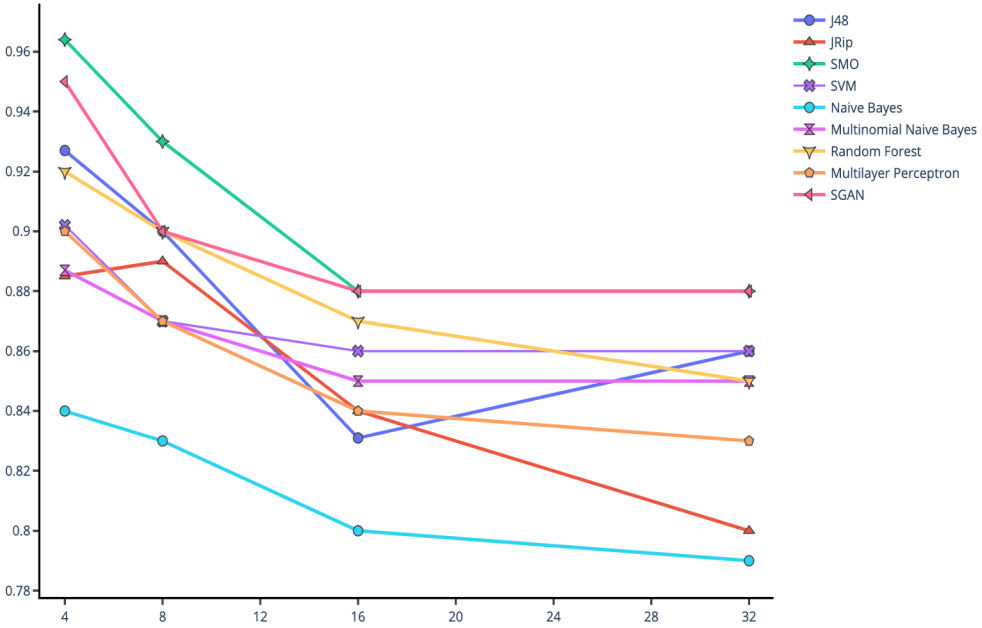


n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.925	0.83	0.94	0.9	0.86	0.89	0.92	0.9	0.94
8	0.9	0.89	0.93	0.87	0.85	0.88	0.95	0.89	0.92
6	0.865	0.85	0.875	0.865	0.82	0.87	0.875	0.86	0.89
32	0.85	0.82	0.875	0.86	0.8	0.875	0.85	0.83	0.88



Recall

n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.93	0.89	0.99	0.905	0.83	0.885	0.925	0.92	0.97
8	0.9	0.89	0.93	0.87	0.8	0.86	0.9	0.85	0.95
6	0.8	0.83	0.89	0.86	0.785	0.835	0.88	0.82	0.93
32	0.87	0.79	0.89	0.86	0.78	0.83	0.865	0.84	0.93

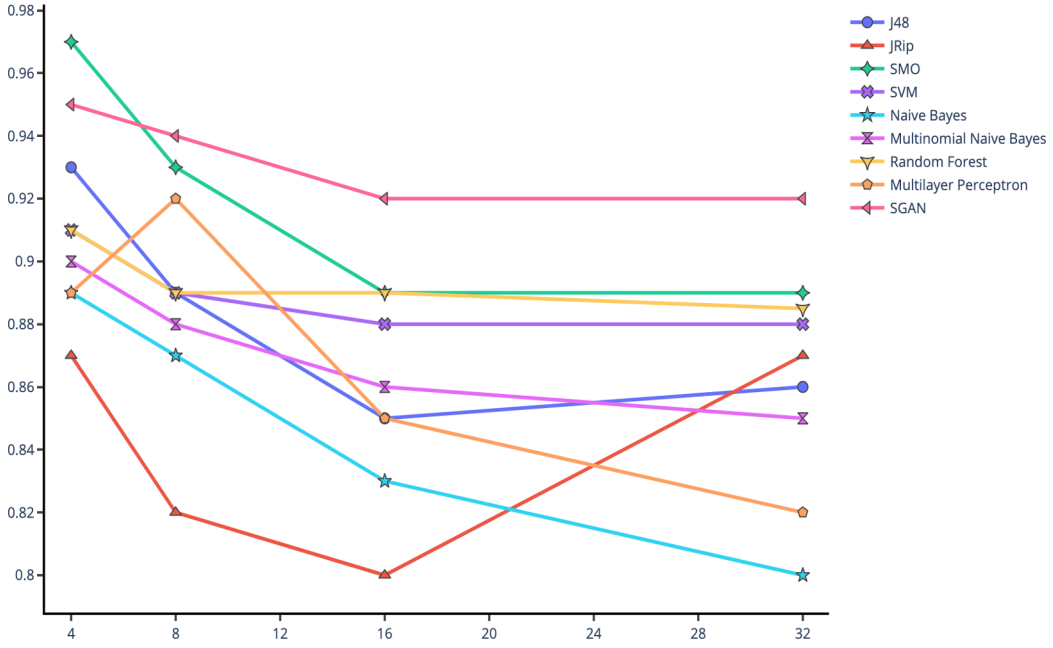


F-Score

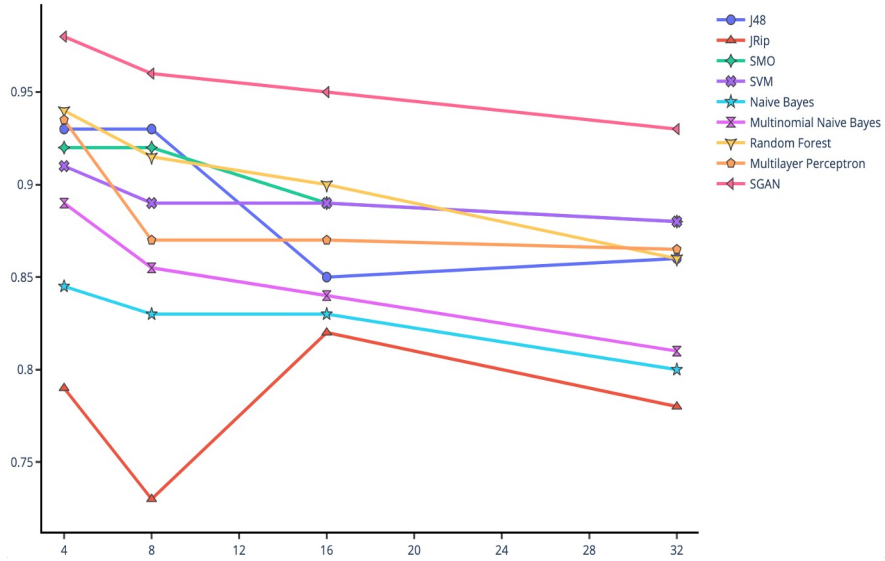
n-GRAMS	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.927	0.885	0.964	0.902	0.84	0.887	0.92	0.9	0.95
8	0.9	0.89	0.93	0.87	0.83	0.87	0.9	0.87	0.9
6	0.83	0.84	0.88	0.86	0.8	0.85	0.87	0.84	0.88
32	0.86	0.8	0.88	0.86	0.79	0.85	0.85	0.83	0.88

Experiment 3: In this experiment we chose 165,000 executables belonging to ten different malware families and one benign family as classified by ClamAV antivirus engine.

Precision

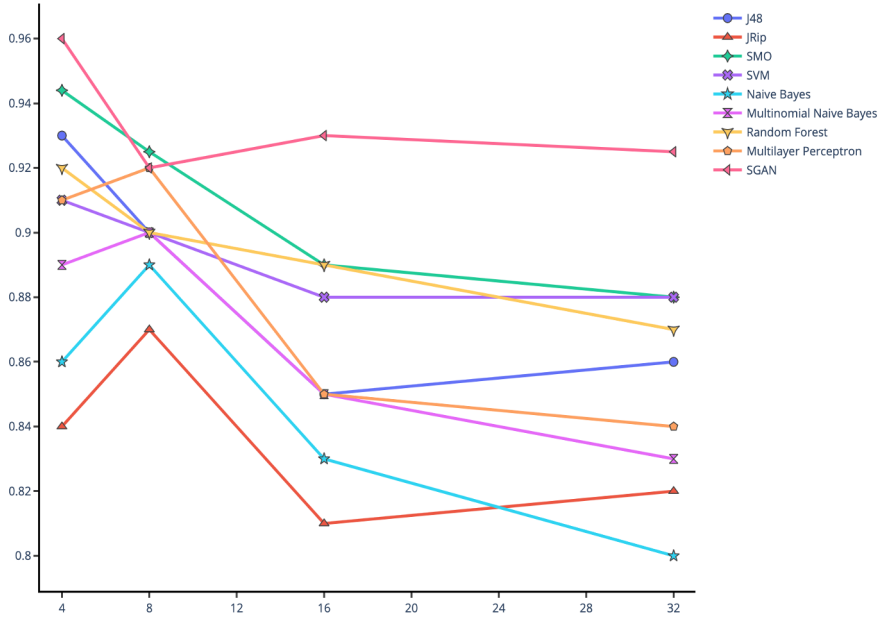


X	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.93	0.87	0.97	0.9	0.89	0.9	0.9	0.89	0.95
8	0.89	0.82	0.93	0.89	0.87	0.88	0.89	0.92	0.94
6	0.85	0.8	0.89	0.88	0.83	0.86	0.89	0.85	0.92
32	0.86	0.87	0.89	0.88	0.8	0.85	0.885	0.82	0.92



Recall

X	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.93	0.79	0.92	0.9	0.845	0.89	0.94	0.935	0.98
8	0.93	0.73	0.92	0.89	0.83	0.855	0.95	0.87	0.96
6	0.85	0.82	0.89	0.89	0.83	0.84	0.9	0.87	0.95
32	0.86	0.78	0.88	0.88	0.8	0.8	0.86	0.865	0.93



F-Score

X	J48	JRIP	SMO	SVM	NAIVE BAYES	MULTINOMIAL NAIVE BAYES	RANDOM FOREST	MULTILAYER PERCEPTRON	SGAN
4	0.93	0.84	0.944	0.9	0.86	0.89	0.92	0.9	0.96
8	0.9	0.87	0.925	0.9	0.89	0.9	0.9	0.92	0.92
6	0.85	0.8	0.89	0.88	0.83	0.85	0.89	0.85	0.93
32	0.86	0.82	0.88	0.88	0.8	0.83	0.87	0.84	0.925

CONCLUSION

- We are able to detect unknown zero day malware variants by using automatically generated signatures to identify patterns in PE-32 executable files with high accuracy.
- Certain algorithms such as SMO, Random Forest and SGAN show a high degree of accuracy
- Algorithms such as Naive Bayes and JRip lack in terms of accuracy.
- We have also observed that the n-gram size of 4 always leads to better accuracy for all the algorithms.

REFERENCES

[1] Asaf Shabtai, Robert Moskovitch et al., "*Detection of malicious code by applying machine learning classifiers on static features: A state-of-the- art survey*", Information Security Technical Report, Volume 14, Issue 1, February 2009, Pages 16-29, ISSN 1363-4127.

[2] Ivan Firdausi, Charles Lim, Alva Erwin,Anto Satriyo Nugroho, "*Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection*", In Proceedings of the 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies

[3] Olivier Henchiri ; Nathalie Japkowicz, "*A Feature Selection and Evaluation Scheme for Computer Virus Detection*", ICDM '06 Proceedings of the Sixth International Conference on Data Mining

[4] Hemant Rathore, Swati Agarwal, Sanjay K. Sahay and Mohit Sewak, "*Malware Detection using Machine learning and Deep learning*", Springer, LNCS, Vol. 11297, pp. 402-411, International Conference on Big Data Analytics, 2018

[5] Sitalakshmi Venkatraman, Mamoun Alazab, R.Vinayakumar, "*A hybrid deep learning image-based analysis for effective malware detection*", 2010 International Conference on Availability, Reliability and Security

THANK YOU