A Tool For Malware And Malicious URL Classification (MAMUC)

Ch.Bhanu Pramod[a], Dr. G.Suresh Reddy[b]

[a] VNR VJIET, Pragathi Nagar , Nizampet (S.O.) , Hyderabad 500090
[b]Ph.D(CSE)Professor & IT Head ,VNRVJIET ,Bachupally,Hyderabad

Corresponding author: CH.BHANU PRAMOD (bhanupramod7@gmail.com)

Abstract: With the rise in use of internet ,also have grown the security concerns associated with it. The most common threats that we encounter on the internet are Malicious URLs and Malware.Traditional solutions for combating these threats, are to build a databases of known sources of trouble or/and using filters to restrict access to resources. This is however not dynamic enough to detect attacks of smart Cyber-criminals. Techniques such as type-squatting, domain-squatting , code obfuscation,etc are hard to identify using conventional methods. Thus to stay one step ahead in the war against Cyber-crime we need to use Machine and Deep learning enabled methods in our defense equipment .

The tool Malware and Malicious URL Classifier (MAMUC) is a unique tool which has 3 features. The first feature helps us to identify if a given URL is malicious or benign. This classifier runs on a novel Malben dateset. The second feature is used to covert a malware sample from byte-code to its corresponding malware image. This called as malware visualization,which is a strategy used in static malware inspection. The third feature of the tool is a malware classifier. Once malware is converted to an image ,its analysis is simplified to a case of image classification. This part identifies to which family an input malware is most closely associated with. Once the malware family is identified it is easier to deal with it. The novelty we have employed in this classifier is the use of Dual channel CNN(DCCNN) algorithm for malware classification.

Keywords: Malware visualization, URL classification , Malware classification,Deep Learning, DCCNN .

---

1. Introduction

Two major security concerns while using the internet are Malicious URLs and Malware. Malicious URLs are those URLs that are either deceptive or contain unsafe content for the user .On the other hand Malware is any program that deteriorates or hinders the performance of the user's system and user's experience.Most of the new Malware is prepared using certain generation tools which means the malware creator does not need to invest much time and effort to create it . In 2020 alone nearly 120.94 Million malware samples were recorded by AV test.This is a growth of 7% compared to 2019. The Clop ransomware was one of the biggest malware threats in 2020.The attackers demanded an amount of more than 20+ Million Dollars to restore services, from the victim.

While combating malware using traditional techniques we thus encounter two problems - one is the quantity of malware and two , the complexity of it. Most new malware has some resemblance to older variants. Identifying and classifying malware into known families is thus very useful as it helps to estimate its behaviour and identify the countermeasures. One breakthrough in the area of malware analysis is Malware Visualization. This is a technique in which malware is converted to a Malware image.Another vector exploited by cyber-criminals to disrupt services and infect systems is through the use Malicious URLs. Malicious URLs try to trick the user into surrendering sensitive information like credit card information and login credential by imitating original websites and the URL that is used to access it. Also URLs which leads to any content that is socially unacceptable or contains potentially dangerous content that user is not expecting, can be considered as malicious URL.

In this paper we describe our tool MAMUC , that tries to address these issues .It is built with 3 capabilities . One, to classify URL as either "SAFE" or "UNSAFE" ; two, a feature to convert malware to images and ;three, a multi-class Malware classifier.

We have used 3 datasets for building our tool. A novel Malben data-set used for URL classification ,secondly the Malimg data-set for Malware classifier and finally the BIG 2015 Microsoft malware challenge data-set for random malware samples. The the remainder of the paper is divided into 7 parts. In section 2 some previous works in these domains are described. In section 3 we discuss the steps involved in converting malware samples to malware images, section 4 is a discussion on Malware classification ,section 5 describes the functioning of the URL classifier and section 6 describes some experiments we did with other classifier models . Finally section 7 contains some concluding remarks and section 8 contains some acknowledgments .

2. Significance Of The Study

Every organization is at the constant threat cyber attacks. Due to the monetary investment required and also because of the complexity of the technology , it is difficult to have effective cyber security solutions. Many cyber security solutions are available yet they are not dynamic enough to adapt to a constantly evolving cyber crime.Among the different vectors of cyber crime , Phishing/malicious URLs and Malware are the most actively used.By using a suitable tools it is possible to overcome all these challenges. MAMUC is a tool designed keeping in mind all these challenges.Users of this tool can intuitively understand how the tool functions.The use of machine learning and deep learning algorithms makes this tool especially sophisticated yet fast when it comes to execution time.

3. Review Of Related Studies

URL Classification

Machine learning has been employed previously for classifying phishing URLs. The Phishtank data-set is popular in this domain . Phishing URLs are those which try to mimic genuine websites and extract sensitive information from the users , like credit card details and passwords.Jeeva et al.(2016)[7] have done classification of URLs but they have used heuristic rules which is only slightly better than blacklisting, Li, Yukun, et al. [11] have exploited the HTML features but its very easy to scrape HTML content of a genuine website and create a malicious website . Patgiri, Ripon, et al[13] have tried SVM and Random forest techniques but their accuracy was not able enter the higher 90%s .The features that sets apart the URL classifier in MAMUC are its speed of execution and convenient user interface , and the accuracy which lies around 99%.

Malware Classification

Different authors have come up with different strategies for malware analysis. One of the advantages of using deep learning models in classification problems is that they are capable of identifying features automatically that we might not be able specify using machine learning models. That is the model interprets features automatically which we might not be able to specify as a feature that must be looked into. Kalash et al.[5] have tried using CNN algorithm and got an accuracy of 98.52 on Malimg data-set , Drew et al.[6] have used gene sequencing techniques to detect malware . While some authors have worked on .asm files like Lee et al.[14] some others have used the .bytes files. The novelty of this project is, we have used DCCNN (dual channel CNN) for malware classification using the malimg data-set.

4. Objectives Of The Study

- To build a simple tool for static malware analysis and URL classification
- To test the performance of DCCNN in Malware Image classification
- To provide a convenient user interface for converting malware to images

5. Hypotheses Of The Study

- DCCNN showed good results in image classification of unbalanced data-set. It was expected that it should be able to classify malware from highly imbalanced Malimg data-set effectively
- Deep learning should be capable of providing good results in Malware classification , and should be able to provide promising solution to problems such as code obfuscation
- The tool should be lightweight and fast

6. About the Data-sets

Malben data-set

For the purpose of URL classification , we have prepared a novel Malben data-set. It contains 450176 URLs and the features of each URL like number of characters, presence of SSL/TLS , number of vowels to the total number of characters ratio, and so on. Of these 266791 are malicious and 183385 are benign URLs. Thr contents of the data-set can be seen in figure 8. The URLs are labeled 0 for genuine and 1 for malicious .

Malimg data-set

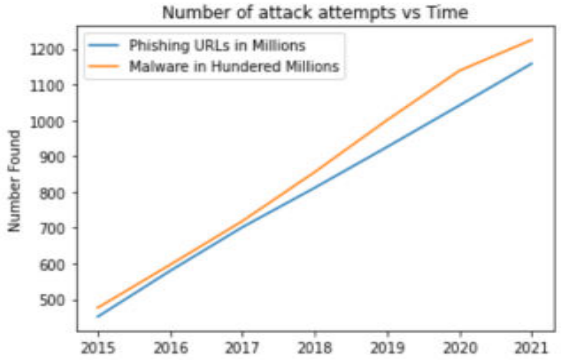
The data-set we have used (Malimg) in our Malware classifier contains malware images as shown in the Figure 4 , that belong to different families of malware. This includes categories of Worms,Trojans ,Trojan Downloaders, Backdoors , Dialers,and so on. On the whole there are malware samples belonging to 25 classes . The malware images size varies according to the size of the malware. The class Allaple.A has the highest number of samples which is nearly 33% of the data-set. Some other families like Skintrim.N contribute only 80 images out of the total 9340 images, which is less than 1%. Thus Malimg is a highly imbalanced data-set which tests the effectiveness of the classifier.

7. Converting Malware to Images

7.1. Theoretical Explanation

Malware analysis can be broadly be classified as static an dynamic.Static analysis works by trying to infer features present in the malware executable file that could possibly suggest intent behind the program. Dynamic analysis which is more resource intensive works by running the malware in controlled conditions. Its behaviour is observed and inferences are drawn.In static analysis Malware Image classification is a recent development.It is a simple and effective technique in which malware samples are expressed as images and they subject to image classification. In this section we discuss the process of converting converting malware to an image.

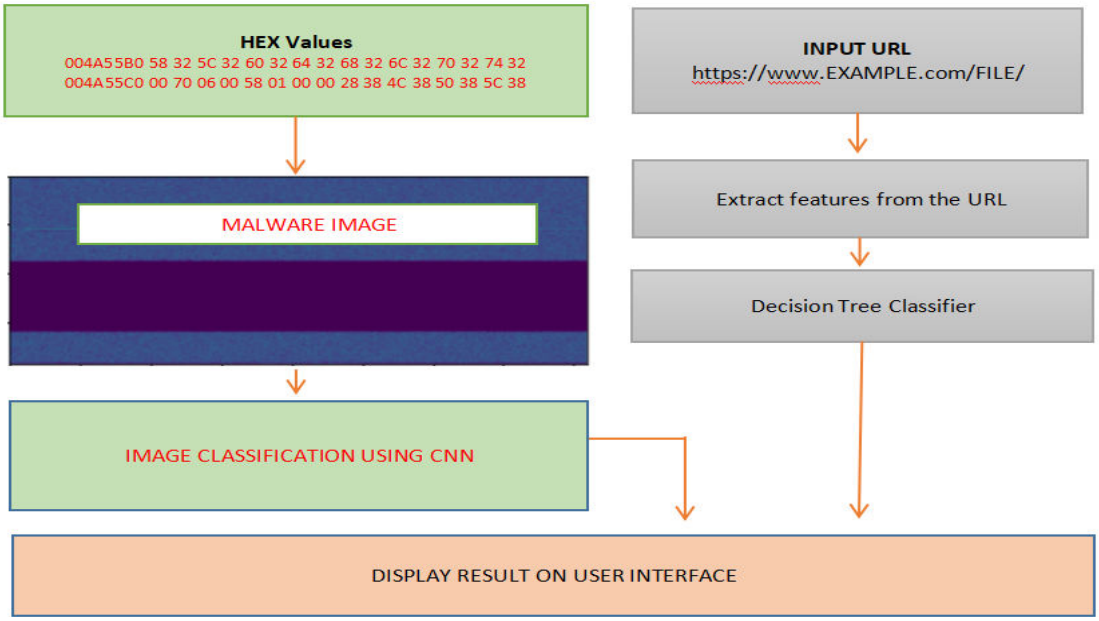
Figure.1 Growth of Cyber-attack with time(Left) and Malware sample as .bytes code(Right)




```
004A5530 44 35 48 35 54 35 64 35 84 35 90 35 94 35 B4 35
004A5540 00 30 0C 30 18 30 30 30 34 30 54 30 58 30 78 30
004A5550 7C 30 9C 30 B0 30 A0 30 C0 30 E0 30 EC 30 F8 30
004A5560 08 31 14 31 24 31 44 31 50 31 54 31 74 31 78 31
004A5570 98 31 9C 31 BC 31 D0 31 E8 31 EC 31 F0 31 F4 31
004A5580 F8 31 FC 31 00 32 04 32 08 32 0C 32 10 32 14 32
004A5590 18 32 1C 32 20 32 24 32 28 32 2C 32 30 32 34 32
004A55A0 38 32 3C 32 40 32 44 32 48 32 4C 32 50 32 54 32
004A55B0 58 32 5C 32 60 32 64 32 68 32 6C 32 70 32 74 32
004A55C0 00 70 06 00 58 01 00 00 28 38 4C 38 50 38 5C 38
004A55D0 60 38 64 38 70 38 74 38 9C 38 78 38 C0 38 C8 38
004A55E0 98 38 BC 38 E0 38 5C 3A 60 3A 64 3A 68 3A 6C 3A
004A55F0 74 3A 8C 3A B8 3A C0 3A A4 3A A8 3A AC 3A B0 3A
004A5600 D8 3A 4C 3B 6C 3B 94 3E A0 3E A4 3E B0 3E C4 3E
004A5610 E8 3E A8 3F CC 3F E8 3F 90 34 94 34 98 34 9C 34
004A5620 A0 34 A4 34 A8 34 AC 34 B0 34 B4 34 B8 34 BC 34
004A5630 C0 34 C4 34 C8 34 CC 34 D0 34 D4 34 D8 34 DC 34
004A5640 E0 34 E4 34 E8 34 EC 34 F0 34 F4 34 F8 34 FC 34
004A5650 00 35 04 35 08 35 0C 35 10 35 14 35 18 35 54 35
004A5660 5C 35 64 35 6C 35 74 35 7C 35 84 35 8C 35 94 35
004A5670 9C 35 A4 35 AC 35 B4 35 BC 35 C4 35 CC 35 D4 35
004A5680 DC 35 E4 35 EC 35 F4 35 FC 35 60 36 64 36 70 36
004A5690 74 36 78 36 08 38 2C 38 08 30 28 30 F4 30 14 31
004A56A0 24 31 28 31 2C 31 30 31 4C 31 50 31 54 31 58 31
```

After normalizing malware images we subject them to classification using the Malimg data-set trained classifier. The tool contains a user interface through which the user can readily generate the image of any malware sample.

Figure.2 High level explanation of MAMUC



7.2. Conversion Process

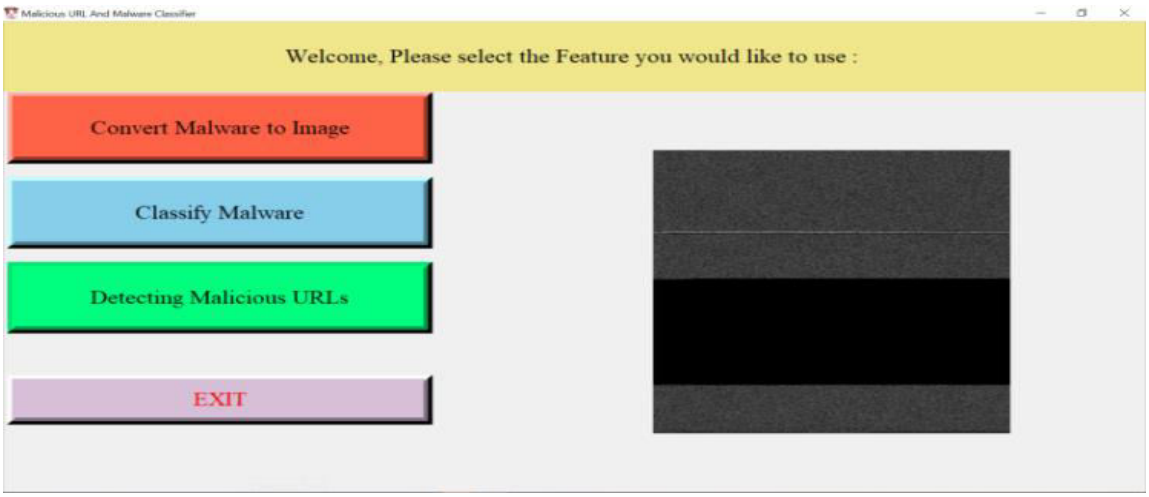
The transformation that is applied to convert malware to image is as follows :-

1. Malware samples in the .bytes format are a string of hexadecimal values.Each line in the malware .Figure 1 depicts the contents of malware sample .bytes code

2. Each line in the .bytes file is split further to extract these hex values ,using which we can form an array of hex values.
3. If the element is not a number we assign a value 0 in its place to the transformed array ; otherwise we assign a value which is the decimal number corresponding to the hexadecimal notation in the transformed array
4. Next the shape of the malware image array is determined based on the dimensions of the transformed array. The transformed array is padded with a sequence of zeroes based on the identified dimensions.
5. The image is created using this array and with the elements determining the pixel values .This results in a malware image. The malware image is displayed on the user interface as in Figure 3.

Each images is a combination of pixels. Generally pixels are made of 3 colors Red,Green and Blue. The values assigned to these colors in a single pixel determines the intensity with which each color fires. The pixel produces that color which is the overall combination of the above 3 colors intensity. The values of pixel can lie in between 0 and 255. 0 indicates black and 255 indicates white. So any values in between represents a specific color. Malware images come under the category of Grayscale images. This means no color is taken into account but only the intensity is considered. So the entire image is combination of different shades of Grey ranging from 0(min intensity) to 255 (highest intensity) .The same process can be repeated for all malware samples in the BIG 2015 data-set. Since this is a very resource intensive process we decided to use the Malimg data-set which already contains the required converted malware images. This data-set is used to train our model. After conversion the malware image can viewed in the user interface of the MAMUC tool as given in Figure 3.

Figure.3 Malware Image result as seen in MAMUC UI



8. Malware Classification

8.1. Working Of The Classifier

Convolution is the process of passing inputs values through filters then multiplying and assigning weights .The result of this operation is always a scalar value and the filter size is always smaller than the input size. Both the models CNN and DCCNN work using convolution . For the CNN mode size of all input images is adjusted to 64 x 64 before using them in the model training. After they pass through the first convolution layer a dropout layer is used, which randomly drops features detected from first layer. The second convolution layer works on these inputs and a maxpooling layer picks the largest element from 4 cells in each 2x2 sub-matrix from its input, and passes to the next convolution layer.The same process of convolution and maxpooling is repeated one more time . After dropping some more features we use a flatten layer which convert the data to a one dimensional array. Dense layers that follow contain fully connected Neural network layers.

Figure.4 Contents of the Malimg data-set(on left) and architecture of DCCNN (on right)

Malware Type	Malware Family	Number of Samples
Worm	Allaple.L	1591
	Allaple.A	2949
	Yuner.A	800
	VB.AT	408
PWS	Lolyda.AA1	213
	Lolyda.AA2	184
	Lolyda.AA3	123
	Lolyda.AT	159
Trojan	C2Lop.P	146
	C2Lop.gen!g	200
	Skintrim.N	80
	Alueron.gen!J	198
	Malex.gen!J	136
TDownloader	Swizzor.gen!I	132
	Swizzor.gen!E	128
	Wintrim.BX	97
	Dontovo.A	162
	Obfuscator.AD	142
Backdoor	Agent.FYI	116
	Rbot!gen	158
Dialer	Adialer.C	122
	Dialplatform.B	177
	Instantaccess	431
Worm:AutoIT	Autorun.K	106
Rogue	Fakerean	381
Total		9339

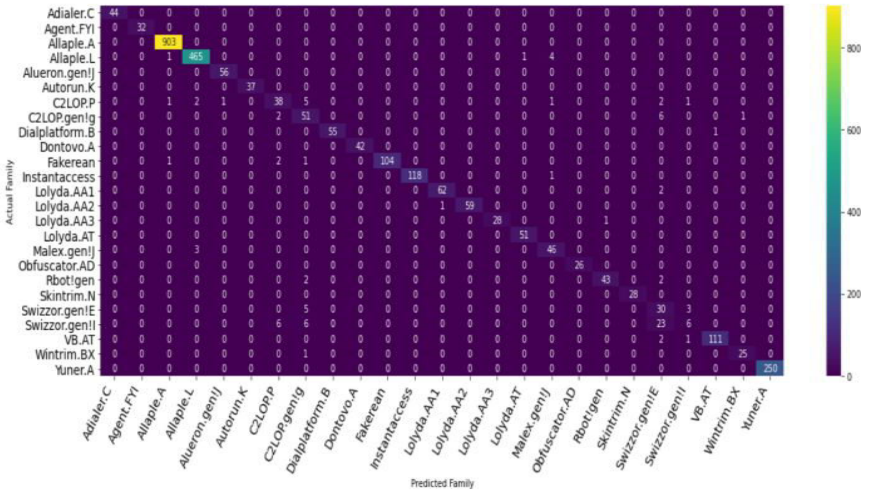
The architecture of the DCCNN model is as shown in Figure 4 (right). DCCNN model has two channels in which input is split into two channels of convolution. The results of both the channels are combined (merged) and then labeling is done. The accuracy of the CNN model was seen to be exactly 94.34%.This is an error rate of 5.66%.Generally when accuracy exceeds 94% aiming for higher accuracy using different techniques like transfer learning requires a trade-of in terms of computation time. Therefore to keep tool light weight we have not used any such techniques. However we have tried to exploit the DCCNN [1] model in the tool.The DCCNN [1] model resulted in a slightly better accuracy than the CNN model at 96.71%. .The classification result window in the UI displays the classification results from multiple models.

Let us consider that D represents a data-set with different malware families.D ={dij} represents the data-set. Here i=1,2,.. represents the family of the malware and j =1,2,3,.. represents the sample in the particular family. The objective of classification is to accept an input malware image ,which might or might not be present in the data-set and identify to which class/family the malware bears highest resemblance with. The tool has the feature of converting malware sample to an image as discussed earlier . Now the same image can be subject to classification in the second feature.

8.2. A Discussion On Malware Classifier Results

One distinguishable feature that was observed in the use of DCCNN[1] was that it was able to differentiate between Autorun.K and Yuner.A families. We have tested other models like AlexNet , VGG16 and VGG19 on the data-set as well. The results of which will be discussed in section 6. None of these algorithms was able to match the precision of DCCNN [1]on the Malimg data-set.Autorun.K and Yuner.A both come under different kinds of Worms families. Also the number of samples in Autorun.K is limited in data-set . As a result of these two reasons most models are unable to distinguish between the two families and wrongly classify the former as the latter.However DCCNN classifies these malware correctly as can be seen from the Confusion matrix in Figure 5.

Figure.5 Confusion Matrix of DCCNN algorithm of Malware classifier.Presence of most of the values along the diagonal of the matrix indicates high level of precision.



Accuracy scores are not always a good indicator of the algorithm performance. Only when we examine the precision and recall can we identify the real usefulness of the model. In multi-class classification we need to examine these parameters for each class , if required we can do an average to estimate the overall behaviour.

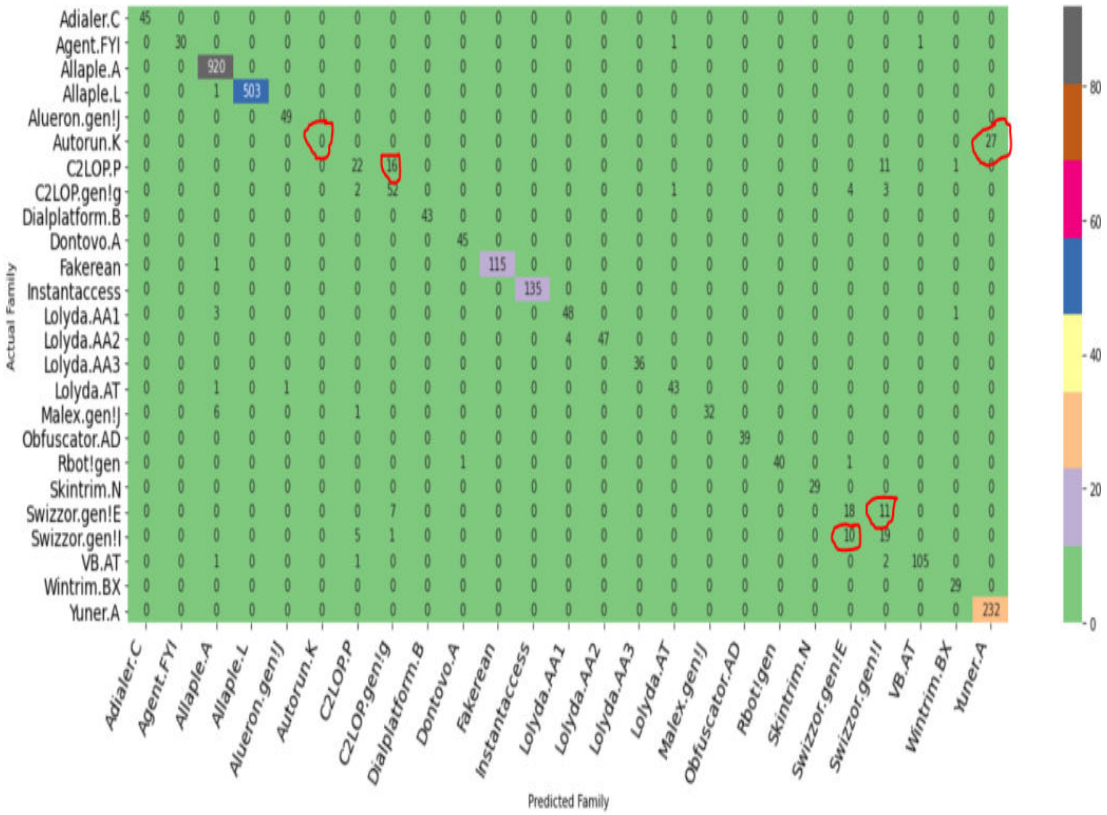
8.3. DCCNN

The lowest precision was found for Swizzor.gen!I at 28.57%. The precision of most classes remained in between 97 and 100% ,However for C2LOP.P,C2LOP.gen!g, Swizzor.gen!E the precision dipped below 75%. This is understandable as each of these families contributes less than 3% to the data-set and also since the first two ; Swizzor.gen!I and Swizzor.gen!E being related families they could be wrongly classified. Interestingly even the lot of families touch 100% precision Allaple.L which is nearly 33% of the data-set does not . Recall values show the similar pattern for DCCNN model. Specificity which is the fraction of true negatives classified as negative by the model , was found to be greater than 99% throughout.

8.4. AlexNet

C2LOP.gen!g and Wintrim.BX were the least performers with AlexNet precision at 23.07% and 20.68% precision. Worth noting is that the disparity of precision between related families like Swizzor.gen!I and Swizzor.gen!E ,and C2LOP.P and C2LOP.gen!g was higher for AlexNet - nearly a 40 % difference . This could indicate that AlexNet is able to extract those features which DCCNN is missing out on in certain cases. While recall showed the same pattern , nothing notable could be found in specificity.

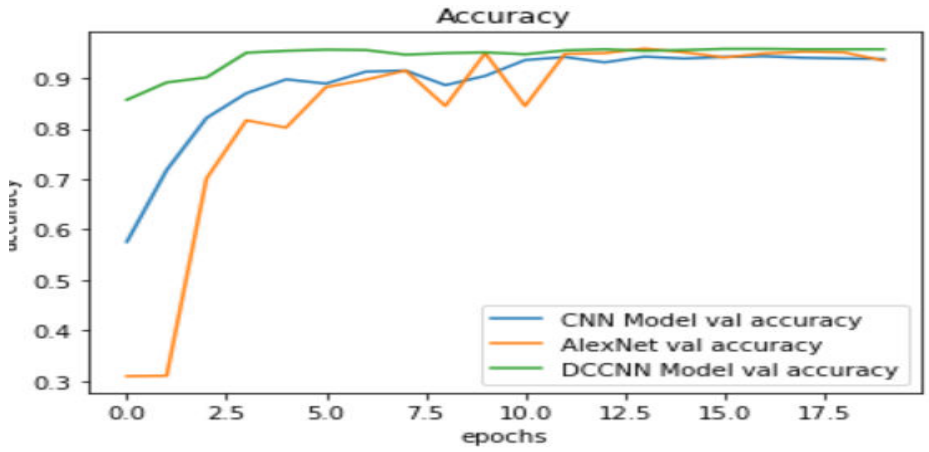
Figure.6 Confusion matrix of CNN wrongly classifying Autorun.K. as Yuner.A.



8.5. CNN

Apart from Swizzor.gen!I at 25% all other classes showed at least 65% recall. Yuner A. showed 100% recall in all models which is worth noting. But specificity of Yuner A. was the least for CNN ,whereas for other classes it exceeded 99%.

Figure.7 Val accuracy vs epochs for all models used in the project



9. Malicious URL Classifier

In this section we are going to examine the final part our tool MAMUC. This part deals with URL classification and uses the decision tree algorithm. Decision tree algorithm works by examining historical behaviour and decisions taken when certain parameters have certain values. Then based on this it determines the most - likely outcome that will occur for the current input provided.

Our definition of malicious URLs includes those URLs which lead to websites that contain abusive, terrorism related , hostile nations based , phishing , adult content, pirated content, fake websites,social media, to name a few. In a way ,websites that enhance the productivity of the average user have been tagged ‘Safe’ and those that hamper it as ‘Unsafe’ . Since no data-set exists to address all these issues at once , so we created a custom data-set that would power our classifier.

For our classifier we have given a novel Malben data-set as input. The procedure followed for creating this data-set is as follows -

1. We first took the DMOZ data-set from Kaggle. From this data-set we removed all the columns except the URLs column.
2. To the above data-set we appended the famous Phishtank
3. data-set which contains a few thousand known phishing URLs. Again we removed all the content except for the URLs part.
4. By themselves the two data-set do not provide any conclusive information sufficient to classify a URL as malicious or not as we have extended the definition of malicious URLs beyond Phishing .
5. The next step is to apply transforms to the URL and extract features that could give us some clues that the URL is malicious. This includes-
   - Presence of SSL/TLS (a)
   - Top level domain and Number of subdomains(b)
   - Path length (c)
   - Use of free web-hosting services(d)
   - Presence of special character in the URL,presence of number and hyphens,presence of repeated characters.(e)

- Ratio of number of vowels to total length of URL and character repeats(f)

Any modern legitimate website is expected to have a certificate (a).Here (b) is for detecting those URLs which try to trick the user by replacing say a '.com' with a '.net' which is called Domain squatting . Path length for safe websites is generally limited 20-30 chars(c) . Any genuine commercial website would never use free host as it degrades the brand value (d).(e) helps to identify common website frauds. (f) to are for identifying those URLs which either replace or insert character which the user might not notice when they read it quickly.

6. After extracting the features of the data-set as above; going through each URL in the data-set, we labeled them as either 'safe' or 'unsafe' according the definition of malicious that was previously given.
7. This column was marked as the result column.
8. Finally since Decision Tree classifier can only process numbers - true values in the data-set were marked as 0 and false values as 1 for all features . If the feature was represented by a number- it was left as it is.
9. In the results column 0 indicates 'Safe' URL and 1 indicates URL with potentially dangerous content - 'Unsafe.The data-set appears like in Figure 8.

Figure.8 Inside the Malben data-set

path	result	ssl	freehost	domain	contains_ni	hyphen	subdomain	slashes	path_lengtl	cha
https://www.google.com	0	0	1	0	1	0	2	2	22	
https://www.youtube.com	0	0	1	0	1	0	2	2	23	
https://www.facebook.com	1	0	1	0	1	0	2	2	24	
https://www.baidu.com	0	0	1	0	1	0	2	2	21	
https://www.wikipedia.org	0	0	1	1	1	0	2	2	25	
https://www.reddit.com	1	0	1	0	1	0	2	2	22	
https://www.yahoo.com	0	0	1	0	1	0	2	2	21	
https://www.google.co.in	0	0	1	1	1	0	3	3	24	
https://www.qq.com	0	0	1	0	1	0	2	2	18	
https://www.amazon.com	0	0	1	0	1	0	2	2	22	
https://www.taobao.com	0	0	1	0	1	0	2	2	22	

Decision tree algorithm which comes under category of supervised algorithms works on the basis splitting strategy. The strategy in a iterative way tries to identify the information gain and entropy at each node and then further splits the data. Entropy is the measure of randomness present in the data and Information gain is the the additional information that can be inferred by taking an action , in other words the extent to which entropy is reduced by taking an action, in this context splitting data. The formula for Information is given below.

Information gain = Entropy before split - Entropy After split

The algorithm examines the decisions taken (result) for a historically for set of inputs in training data and then tries to infer and approximate ,the most likely outcome of provided data. We have used Gini index in the Decision tree, it works by examining the impurity present in the data after each split and tries to minimize it. The accuracy score of the classifier obtained was - 0.996653

9.1 Algorithm Performance Metrics

For analyzing the performance of a classifier the metrics in this section can be used.

Before that we need to understand the following terms-

- True-positive (TrPo)- actual and predicted outcomes both are positive
- True-negative (TrNe)- actual and predicted outcomes are both negative
- False-positive(FaPo) - actual outcome is false but predicted outcome is true

- False-negative(FaNe) - actual outcome is True but predicted outcome is false

Precision is the measure of accuracy of positive predictions and Recall is the measure of percent of positive cases that were captured.

$$Precision = TrPo/(TrPo + FaPo)$$

$$Recall = TrPo/(TrPo+FaNe)$$

F1score is the measure of percentage of positive predictions that were correct. It is obtained from the harmonic mean of precision and recall.

Table.1. Confusion matrix of URL classifier

	Predicted UNSAFE	**Predicted SAFE**
Actual UNSAFE	*79670*	*12*
Actual SAFE	*440*	*54931*

9.2 Results obtained :

From the confusion matrix the following values can be obtained :

Precision =0.9997815918315345

Recall =0.9920536020660634

F1-score = 0.9959026052903529

Figure.9 Result windows of URL classification as seen in the MAMUC UI. Left window in the figure shows SAFE, and right window shows result for Malicious URLs.

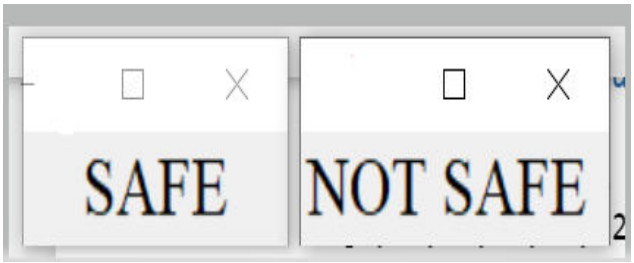


Table.2. Accuracy comparison table in malware classification.

Algorithm	**Val Accuracy**	**Can distinguish Autorun.K from Yuner.A ?**
CNN	*94.34*	*No*
DCCNN	*96.71*	*Yes*
AlexNet	*94.36*	*Yes*
VGG16	*32.83*	*No*
VGG19	*32.83*	*No*

10. Experiments done with the Malware classifier in MAMUC .

We have tried to train models like VGG16 and VGG19 on unbalanced data-set, but the results were not satisfactory ,not to mention the unfeasible training time of these models.Vgg16 simply cannot be used for malware classification on Malimg data-set directly (without proper sampling) and gives very low accuracy scores. It was observed that all malware samples were being classified as belonging to Allaple.A. family by these two models . This is probably because, bulk of the images in the data-set belong to this class. For this VGG16 was getting accuracy score of 32.83 % in this scenario. This could probably hint at heavy over-fitting as number of samples of Allaple.A family are nearly 30% of the entire data-set. VGG19 produced exactly the same results even though VGG19 exploits around 5 Million more parameter , so both these algorithms are not suitable options for malware classification on raw Malimg data-set. By keeping the number of filters constant at 15 through all layers in VGG16 architecture , the accuracy shot up to 94.39 % ! A similar behaviour can be observed with VGG19.

Though accuracy appears to be only a tad higher than that of other algorithms, the confusion matrix obtained for DCCNN indicates that , the ability to identify to distinguish between close families is higher in it is higher .Consider Swizzor.gen!l and Swizzor.gen!E , most algorithms wrongly classify the latter for the former. But in case of DCCNN this error is negligible.

Another algorithm that was tried was AlexNet ,the accuracy obtained was 97.78 % after 10 epochs with reduced number of filter compared to the original models . The size of the images used in the model are limited to 64 x 64 . Without changing filter size the accuracy changes dramatically with each epoch , again hinting at heavy over-fitting. Among all models AlexNet has the most number of layers.

11. Recommendations

- The tool MAMUC can be used on personal computers for identifying and classifying and suspicious URLs one might come across while surfing the web or while going through emails.
- MAMUC can also be used to convert malware samples to their corresponding images then classifying these images to families of known malware samples. Once the family of Malware sample is identified it is easier to perform malware analysis on the Malware sample
- The simple user interface helps to simplify the process of static Malware analysis

12. Conclusion

In this paper, we have described the basic features of MAMUC. The tool is simple and effective for Malware and URL classification . Though MAMUC is lightweight and fast it was developed with a focus to make classification as user_friendly as possible , yet it gives very good results. Most of the issues with current models of URL classification - like inability to identify free hosting ,domain squatting , URL obfuscation, overlooking HTTPS are addressed by the model that we have proposed for URL classification .The URL classifier has been developed keeping in mind strict censorship needs that would be required by an educational institution ,thus any website that violates the needs of an educational institution will be classified as “UNSAFE”. The precision of the URL classifier is better than other implementations which use Tf-idf Vectorizer and cannot detect the deeper features in URL.Though the tool is able to classify Malware and Malicious URLs it does not take corrective action based on the classification result like an antivirus would do, limiting its scope to diagnosis .Though AlexNet had a higher accuracy than DCCNN , its precision was lower.By allowing custom user input feature the user can convert and classify any random Malware file , not in the data set and use of DCCNN model for malware classification are some more novelties of this project . In future if required we can roll out this project in the form of an application .

Acknowledgments

The author would like to thank Dr.G.Suresh Reddy, for his guidance and suggestions . He is the author’s guide and Head of Dept. ,Information Technology ,VNR VJIET , Hyderabad.

References (APA)

1. Cao, Jianfang, et al. "An improved convolutional neural network algorithm and its application in multilabel image labeling." Computational Intelligence and Neuroscience 2019 (2019).
2. Liu, Ya-shu, et al. "A new learning approach to malware classification using discriminative feature extraction." IEEE Access 7 (2019): 13015-13023.
3. Nataraj, Lakshmanan, et al. "Malware images: visualization and automatic classification." Proceedings of the 8th international symposium on visualization for cyber security. 2011.

4. Singh, Ajay, et al. "Malware classification using image representation." International Symposium on Cyber Security Cryptography and Machine Learning. Springer, Cham, 2019.
5. Kalash, Mahmoud, et al. "Malware classification with deep convolutional neural networks." 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). IEEE, 2018.
6. Drew, Jake, Michael Hahsler, and Tyler Moore. "Polymorphic malware detection using sequence classification methods and ensembles." EURASIP Journal on Information Security 2017.1 (2017): 1-12.
7. Jeeva, S. Carolin, and Elijah Blessing Rajsingh. "Intelligent phishing url detection using association rule mining." Human-centric Computing and Information Sciences 6.1 (2016): 1-19.
8. Nataraj, Lakshmanan, and B. S. Manjunath. "SPAM: Signal processing to analyze malware [applications corner]." IEEE Signal Processing Magazine 33.2 (2016): 105-117.
9. Sahoo, Doyen, Chenghao Liu, and Steven CH Hoi. "Malicious URL detection using machine learning: A survey." arXiv preprint arXiv:1701.07179 (2017).
10. Kumar, Jitendra, et al. "Phishing Website Classification and Detection Using Machine Learning." 2020 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2020.
11. Li, Yukun, et al. "A stacking model using URL and HTML features for phishing webpage detection." Future Generation Computer Systems 94 (2019): 27-39.
12. Rathore, Hemant, et al. "Malware detection using machine learning and deep learning." International Conference on Big Data Analytics. Springer, Cham, 2018.
13. Patgiri, Ripon, et al. "Empirical study on malicious URL detection using machine learning." International Conference on Distributed Computing and Internet Technology. Springer, Cham, 2019.
14. Lee, Young Jun, et al. "Learning binary code with deep learning to detect software weakness." KSII The 9th International Conference on Internet (ICONI) 2017 Symposium. 2017