

Τμήμα Πληροφορικής

**Διπλωματική Εργασία**

**Θέμα : Οπτική Κατηγοριοποίηση κακόβουλου λογισμικού**(Visual Malware Classification)

Δελλής Παρασκευάς - Π15034

Επιβλέπων καθηγητής:Πατσάκης Κωνσταντίνος

Η παραπάνω πτυχιακή εργασία ,για την απόκτηση του προπτυχιακού τίτλου σπουδών, πραγματοποιήθηκε σε συνεργασία και συνεχή επικοινωνία με τον καθηγήτη του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς, Πατσάκη Κωνσταντίνο τον οποίο θα ήθελα να ευχαριστήσω θερμά για την ανάθεση αυτής.

Contents

[**Εισαγωγή** 3](#_Toc103375846)

[**Χαρακτηριστικά Υλοποίησης** 4](#_Toc103375847)

[**Τρόπος Ανάλυσης** 4](#_Toc103375848)

[**Malimg Dataset** 5](#_Toc103375849)

[Χαρακτηριστικά εκτέλεσης 6](#_Toc103375850)

[Εκτέλεση και αποτελέσματα 7](#_Toc103375851)

[**Malware Bazaar dataset** 8](#_Toc103375852)

[**Πρώτη μετατροπή** 9](#_Toc103375853)

[Παράδειγμα πρώτης μετατροπής 9](#_Toc103375854)

[Χαρακτηριστικά εκτέλεσης 10](#_Toc103375855)

[Αποτελέσματα πρώτης εκτέλεσης 12](#_Toc103375856)

[**Δεύτερη μετατροπή** 13](#_Toc103375857)

[Παράδειγμα δεύτερης μετατροπής 13](#_Toc103375858)

[Χαρακτηριστικά εκτέλεσης 14](#_Toc103375859)

[Αποτελέσματα δεύτερης εκτέλεσης 14](#_Toc103375860)

[**Σύνοψη** 15](#_Toc103375861)

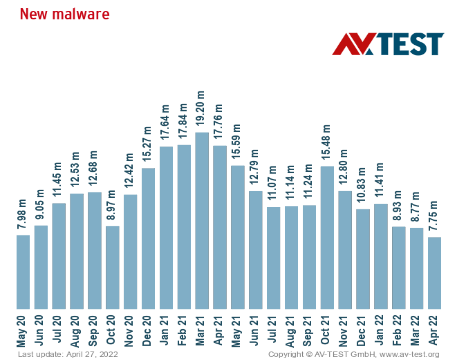
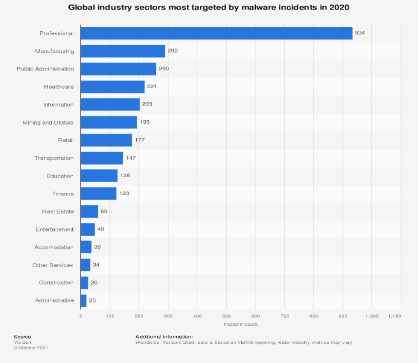
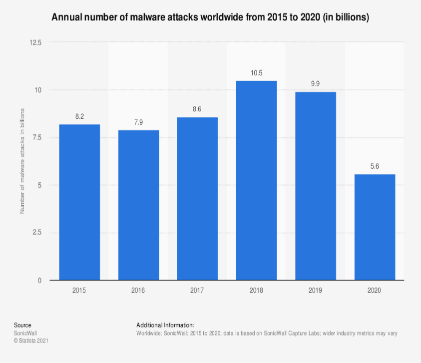
[**Βιβλιογραφία** 16](#_Toc103375862)

# **Εισαγωγή**

Στη σύγχρονη εποχή, εξαιτίας της ραγδαίας εξέλιξης και της εκθετική αύξησης της ανάπτυξης της τεχνολογίας , η χρήση και η αξιοποίηση της δεν αποσκοπεί μόνο στο όφελος της κοινωνίας (π.χ παραπληροφόρηση,καταναλωτικά πρότυπα κλπ.). Ένα χαρακτηριστικό πρόβλημα, το οποίο απασχολεί την επιστήμη της Πληροφορικής( συγκεκριμένα την ασφάλεια των Πληροφοριακών Συστημάτων) ως προς την επίλυση του και μπορεί να βλάψει οποιαδήποτε ηλεκτρονική συσκευή, είναι η ανάπτυξη και η χρήση κακόβουλων λογισμικών( malicious software/malware ή badware) από εγκληματίες του κυβερνοχώρου. Ένα λογισμικό χαρακτηρίζεται κακόβουλο όταν βάσει των προθέσεων του προγραμματιστή το λογισμικό που προκύπτει διαθέτει τις απαιτούμενες εντολές προκειμένου να βλάψει ένα υπολογιστικό σύστημα.

Ειδικότερα, για το παραπάνω πρόβλημα παρατηρούμε:

* Αυτή τη στιγμή υπάρχουν συνολικά περισσότερο από 1 δισεκατομμύριο κακόβουλα λογισμικά, ενώ από το 2013, η εμφάνιση των malware αυξάνεται εκθετικά. Παρόλο που τα τελευταία χρόνια συναντάμε μείωση της αύξησης αυτής, δεν έχει σταματήσει και ακόμα και με τη χρήση antivirus σήμερα υπάρχουν περισσότερα κακόβουλα λογισμικά από ποτέ.
* Περισσότερα απο 450 χιλιάδες καινούρια malwares εντοπίζονται καθημερινά, ενώ μηνιαία το νούμερο ανέρχεται σε παραπάνω απο 12 εκατομμύρια.
* Το 2020, ο συνολικός αριθμός των παραλλαγών των malwares αυξήθηκε κατα 74% σε σχέση με το 2019.
* Οι προαναφερθέντες ιοί κυρίως διαδίδονται μέσω .exe αρχείων (συνολικά το 50 περίπου τοις εκατό όλων των επεκτάσεων αρχείων).
* Στο πρώτο μισό του 2020 οι επιθέσεις με κακόβουλα λογισμικά σε κινητά, ξεπέρασαν των αριθμό των 28 εκατομμυρίων.
* Το 2019, μετά απο επίθεση με ιο που απειλεί οργανώσεις για δημοσίευση ή απώλεια δεδομένων(ransomware attack), ένα ποσοστό 45% αυτών αποφάσισε να πληρώσει τα απαιτούμενα λύτρα, με τις μισές από αυτές να χάνουν και πάλι τα δεδομένα τους. Συνολικά, το 2019 περισσότερα απο 10 δισεκατομμύρια ευρώ δόθηκαν σε λύτρα.



Είναι φανερό, ότι ο σχεδιασμός κακόβουλων λογισμικών και οι επιθέσεις σε υπολογιστικά συστήματα με αυτά, αποτελεί ένα μείζον πρόβλημα του σημερινού κόσμου, με το οποίο έρχονται αντιμέτωποι άνθρωποι,εταιρείες,οργανώσεις κλπ. Αυτό το πρόβλημα είναι δυνατό επίσης να πάρει πολλές διαστάσεις με κυριότερη την οικονομική. Σύμφωνα με τα παραπάνω στατιστικά και το ενδιαφέρον μου στο θέμα, η πτυχιακή εργασία συντάχθηκε ως μία προσπάθεια να γίνει δυνατή η οπτική κατηγοριοποίηση των κακόβουλων λογισμικών.

# **Χαρακτηριστικά Υλοποίησης**

Περιβάλλον Ανάπτυξης : Windows, Virtual Machine Kali Linux

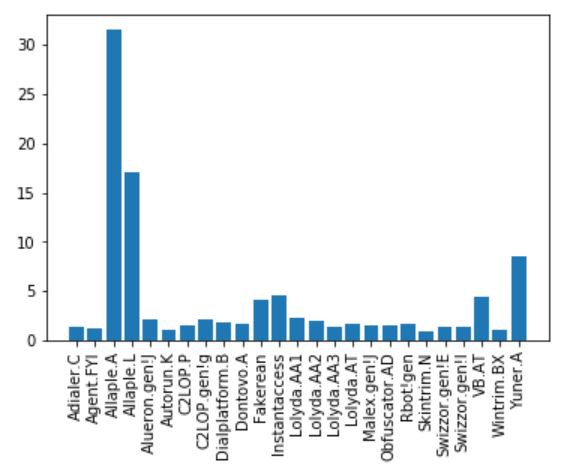
Γλώσσες Προγραμματισμού: Python

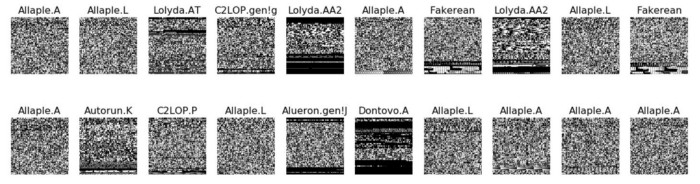
# **Τρόπος Ανάλυσης**

Η δυναμική ανάλυση προγράμματος είναι ανάλυση λογισμικού που γίνεται με την εκτέλεση των προγραμμάτων που προκύπτουν από αυτό το σύστημα λογισμικού σε έναν πραγματικό ή εικονικό επεξεργαστή. Η δυναμική ανάλυση δεν ήταν εφικτή με κακόβουλα λογισμικά εφόσων θα έβλαπταν το σύστημα εκτέλεσης και επομένως πραγματοποιήθηκε στατική, δηλαδή ανάλυση η οποία έγινε χωρίς την εκτέλεση αυτών. Πιο συγκεκριμένα, υλοποιήθηκε μία περισσότερο γενική προσέγγιση του θέματος, δηλαδή μια γρήγορη απεικόνιση των λογισμικών αυτών( τα λογισμικά που επιλέχθηκαν ήταν εκτελέσιμα αρχεία επέκτασης .exe, η οποία ήταν η κυριάρχη για τα κακόβουλα ) ως εικόνες και συγκεκριμένα επέκτασης .png. Η μέθοδος στατικής ανάλυσης ήταν η μετατροπή του δυαδικού περιεχομένου των ακατέργαστων δεδομένων ( raw data ) του κάθε εκτελέσιμου αρχείου σε εικόνα. Οι εικόνες στη συνέχεια έγιναν η είσοδος για ένα συνελικτικό νευρωνικό δίκτυο ή για κάποιον άλλο αλγόριθμο ταξινόμησης εικόνων. Τέτοιες έρευνες διεξάγονται με σκοπό την οπτικοποίηση κακόβουλων λογισμικών για την επίτευξη υψηλής απόδοσης ταξινόμησης και χαμηλού χρόνου.

# **Malimg Dataset**

Τo Malimg dataset αποτελεί ένα μη ισορροπημένο σύνολο δεδομένων από 9339 εικόνες κακόβουλων λογισμικών που ανήκουν σε 25 διαφορετικές οικογένειες/κλάσεις. Αρχικά, έγινε λήψη των δεδομένων και ανάλυση αυτών. Στόχος ήταν η πολυταξική ταξινόμηση των malware, δηλαδή το νευρωνικό δίκτυο που θα δημιουργηθεί, να είναι ικανό χάρη στην εικόνα να προβλέψει σε ποια οικογένεια ανήκει το κάθε κακόβουλο λογισμικό. Το malimg dataset επιλέχθηκε σε πρωταρχικό στάδιο επειδή ήταν ήδη γνωστό και δουλεμένο, για τον έλεγχο και τη σύγκριση των αποτελεσμάτων.





## Χαρακτηριστικά εκτέλεσης

Μοντέλα :

**keras.Sequential ( 10 iterations ) :** Το μοντέλο Sequential αποτελεί μία γραμμική στοίβα στρωμάτων. Στην κλάσση Sequential δημιουργούνται και προστίθενται επίπεδα μοντέλου.

Βασικές πρόσθετες βιβλιοθήκες:

**Sklean.model\_selection.train\_test\_split**

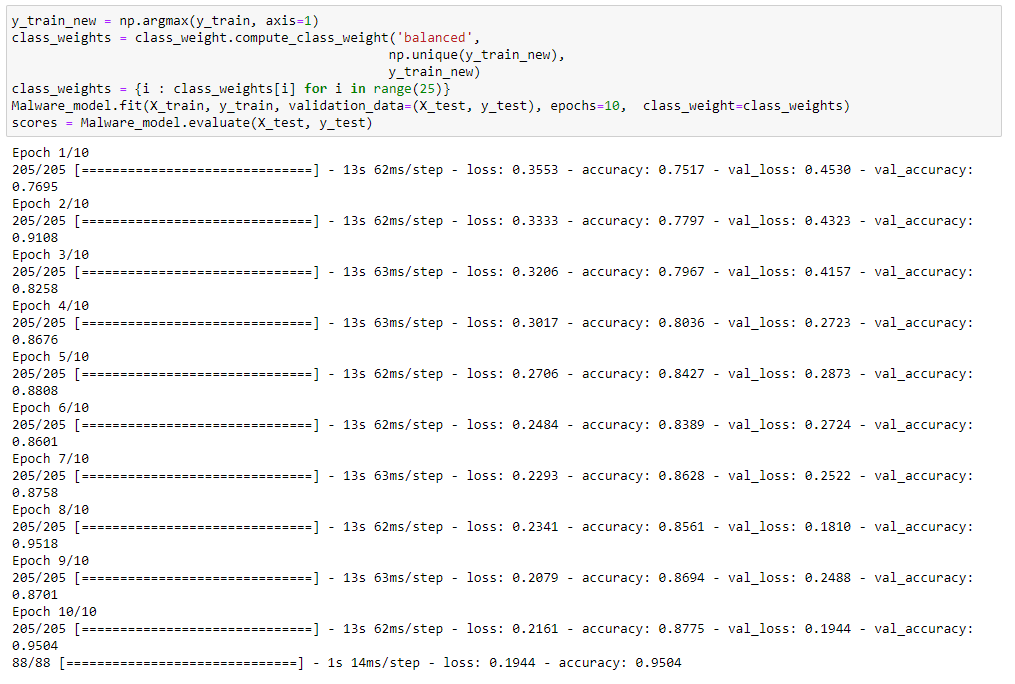
**Sklearn.utils.class\_weight**

Για την εκτέλεση των μοντέλων ακολούθησαν οι εξής προετοιμασίες:

* Κανονικοποίηση των εικόνων (imgs/255.) έτσι ώστε κάθε pixel να έχει παρόμοια κατανομή και το δίκτυο να κάνει τη σύγκλιση ταχύτερη κατά την εκπαίδευση του.
* Χρήση της Sklearn.model\_selection.train\_test\_split για το διαχωρισμό των δεδομένων σε δεδομένα εκπαίδευσης (train data )και δεδομένα δοκιμής ( test data ) ποσοστιαία (70% train 30% test ).
* Με τη βοήθεια της βιβλιοθήκης Sklearn.utils.class\_weight υπολογίστηκε η βαρύτητα κάθε κλάσης του συνόλου.

## Εκτέλεση και αποτελέσματα

Δημιουργήθηκε ενα συνελικτικό νευρωνικό δίκτυο(Convolutional Neural Network) με το μοντέλο Sequential από την βιβλιοθήκη της Python, keras. Επειδή τα δεδομένα δεν ήταν ισορροπημένα, μέσω της βιβλιοθήκης sklearn.class\_weight αποδώθηκε η ανάλογη βαρύτητα σε κάθε οικογένεια εικόνων ξεχωριστά. Τέλος, αφού αναγνωρίστηκαν όλα τα δείγματα και το σύνολο των κατηγοριών τους μέσω της διαδικασίας train\_test\_split, μετά από 10 επαναλήψεις του μοντέλου Sequential, παρατηρήθηκε ένα ποσοστό ακριβείας σχετικά με την ταξινόμηση εικόνων, 95% ( accuracy ).



# **Malware Bazaar dataset**

Στο επόμενο βήμα έγινε λήψη των δεδομένων από το site bazaar.abuse.ch(Malware Bazaar by ABUSE), το οποίο προσφέρει διαμοιρασμό και ανανέωση των καινούριων κακόβουλων λογισμικών με σκοπό την έρευνα και την πρόληψη. Από την ιστοσελίδα αυτή ,μέσω του πιο πρόσφατου αρχείου csv, με τη χρήση των SHA256 hashes κάθε καινούριου εντοπισμένου ιου, πραγματοποιήθηκε λήψη των εκτελέσιμων. Το SHA-256 είναι μια πατενταρισμένη συνάρτηση κατακερματισμού κρυπτογράφησης που εξάγει μια τιμή μήκους 256 bit κι έτσι το SHA-256 hash αποτελεί το κλειδί για την ανάγνωση των δεδομένων. Διατηρήθηκαν μόνο οι οικογένειες ιων με περισσότερα από 100 εκτελέσιμα αρχεία. Τελικά, το καινούριο dataset ηταν αρκετά μεγαλύτερο και περισσοτέρων ομάδων συγκρητικά με το malimg dataset ενώ παρατηρήθηκε ότι δεν ήταν ισορροπημένο.

Ο κώδικας που χρησιμοποιήθηκε για την λήψη του κάθε αρχείου από το αρχείο csv:

**def** download\_file**(**z**,**h**):**

**import** requests

url **=** "https://mb-api.abuse.ch/api/v1/"

params **=** **{**"query"**:** "get\_file"**,** "sha256\_hash"**:** h**}**

r **=** requests**.**post**(**url**,** data**=**params**)**

response **=** r**.**content

**with** **open(**f"D:/samples/'{z}'/{h}.zip"**,** "wb"**)** **as** fout**:**

fout**.**write**(**response**)**

**return** **True**

**with** **open(**"full.csv"**,**"rt"**,**encoding**=**'utf-8'**)** **as** f**:**

csv\_reader **=** csv**.**reader**(**f**,** skipinitialspace**=True)**

**for** line **in** csv\_reader**:**

**if** line**[**7**]** **==** 'application/x-dosexec'**:**

new **=** line**[**8**]**

**if** **not** os**.**path**.**exists**(**r'D:/samples/%a'**%**new**):**

os**.**makedirs**(**r'D:/samples/%a'**%**new**)**

hashes**.**append**(**line**[**1**])**

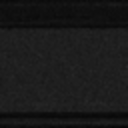
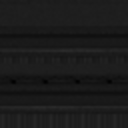
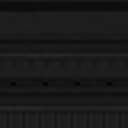
download\_file**(**new**,**line**[**1**])**

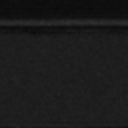
## **Πρώτη μετατροπή**

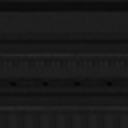
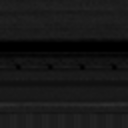
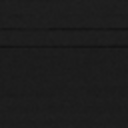
Με τον παραπάνω κώδικα, το σύνολο δεδομένων αποτελούταν από 121.078 αρχεία κακόβουλων λογισμικών, 54ων διαφορετικών κλάσεων ( π.χ AsyncRAT, BitRAT κ.α ). Σε αυτό το στάδιο έγινε η μετατροπή όλων των αρχείων σε εικόνες επέκτασης .png στο λογισμικό των kali Linux χωρίς τη διακινδύνεύση της ασφάλειας του συστήματος. Στην πρώτη μετατροπή το δυαδικό περιεχόμενο κάθε αρχείου αναπαραστήθηκε δεκαεξαδικά (σύμφωνα με το άθρο Malware Classification using Convolutional Neural Networks) και μετατράπηκε σε ασπρόμαυρες εικόνες επέκτασης .png μεγέθους 64x64 pixels.

### Παράδειγμα πρώτης μετατροπής

**9 δείγματα του malware AveMariaRAT**

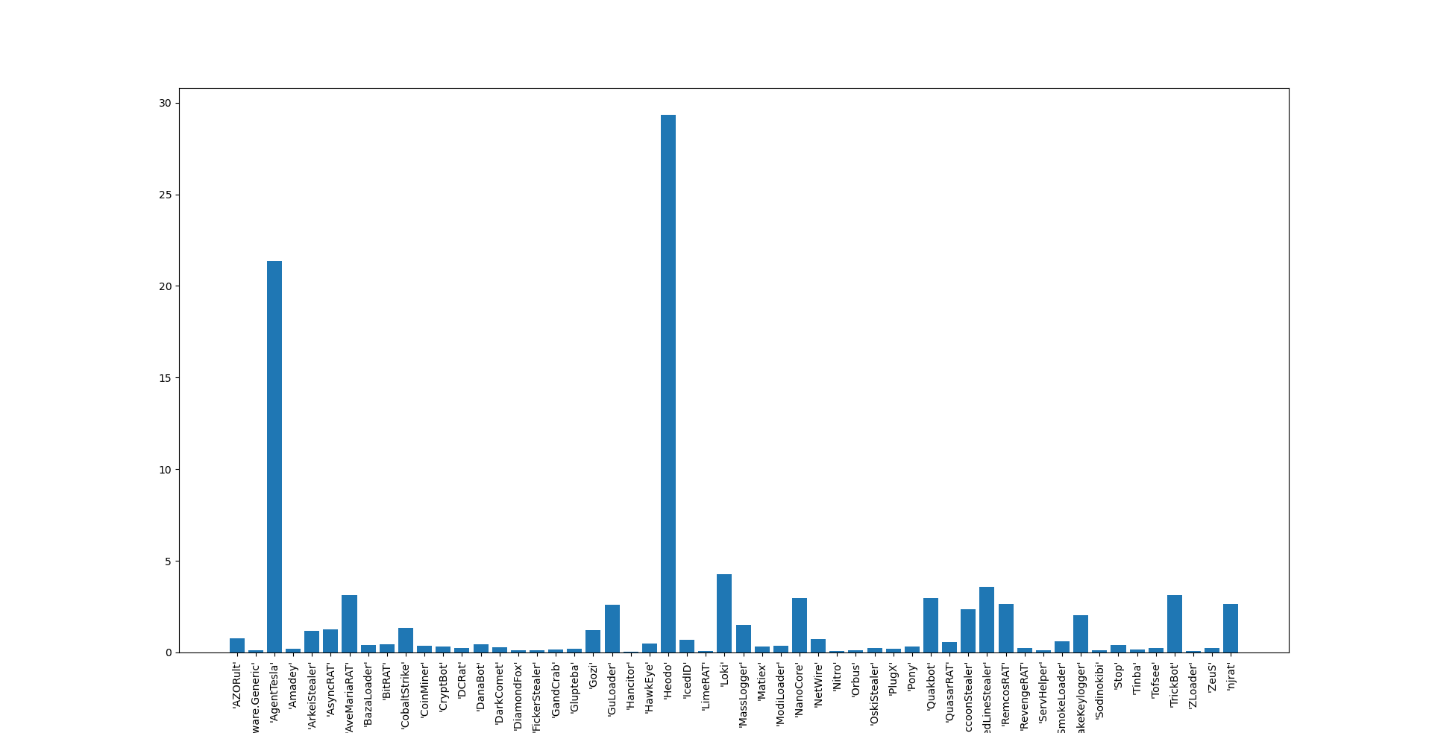
  

### Χαρακτηριστικά εκτέλεσης

Κατανομή των οικογενειών των εικόνων :



Παρατηρείται πολύ υψηλός αριθμός εικόνων για δυο οικογένειες ( Heodo, AgentTesla ).

Μοντέλα :

**keras.Sequential ( 50 iterations ) :** Το μοντέλο Sequential αποτελεί μία γραμμική στοίβα στρωμάτων. Στην κλάσση Sequential δημιουργούνται και προστίθενται επίπεδα μοντέλου.

**xgboost ( 50 iterations )** : Το XGBoost είναι ένας αλγόριθμος μηχανικής μάθησης που βασίζεται σε δέντρα αποφάσεων που χρησιμοποιεί ένα πλαίσιο ενίσχυσης κλίσης.

**sklearn.ensemble.RandomForestClassifier** : Ένα τυχαίο δάσος( random forest ) είναι ένας μετα-εκτιμητής που ταιριάζει σε έναν αριθμό ταξινομητών δέντρων αποφάσεων σε διάφορα υποδείγματα του συνόλου δεδομένων και χρησιμοποιεί τον μέσο όρο για να βελτιώσει την προγνωστική ακρίβεια και τον έλεγχο της υπερπροσαρμογής.

Βασικές πρόσθετες βιβλιοθήκες:

**Sklearn.model\_selection.KFold**

**Sklean.model\_selection.train\_test\_split**

**Sklearn.metrics.classification\_report**

**Sklearn.utils.class\_weight**

Για την εκτέλεση των μοντέλων έγιναν οι εξής προετοιμασίες στο δίκτυο:

* Κανονικοποίηση των εικόνων (imgs/255.) έτσι ώστε κάθε pixel να έχει παρόμοια κατανομή και το δίκτυο να κάνει τη σύγκλιση ταχύτερη όταν εκπαιδεύεται.
* Χρήση είτε του αλγόριθμου Sklearn.model\_selection.KFold είτε του Sklearn.model\_selection.train\_test\_split για το διαχωρισμό των δεδομένων σε δεδομένα εκπαίδευσης (train data )και δεδομένα δοκιμής ( test data ) ποσοστιαία (70% train 30% test ).
* Με τη βοήθεια της βιβλιοθήκης Sklearn.utils.class\_weight υπολογίστηκε τη βαρύτητα κάθε κλάσης.
* Αφαίρεση του one hot encoding και μετασχηματισμός των πινάκων των φωτογραφιών για όσα μοντέλα οι είσοδοι ( input ) τους δεν το επέτρεπαν.
* Προσθήκη της βιβλιοθήκης Sklearn.metrics.classification\_report έτσι ώστε να είναι ορατές οι κύριες μετρήσεις ταξινόμησης:
* Precision : Εμφανίζει ποιο ποσοστό θετικών ταυτοποιήσεων ήταν πραγματικά σωστό.
* Recall: Εμφανίζει ποιο ποσοστό των πραγματικών θετικών προσδιορίστηκε σωστά.
* Accuracy: Η ακρίβεια είναι η αναλογία των αληθών αποτελεσμάτων μεταξύ του συνολικού αριθμού των υποθέσεων που εξετάστηκαν.
* F1 score: Η βαθμολογία F1 ορίζεται ως ο αρμονικός μέσος όρος μεταξύ ακρίβειας( precision ) και ανάκλησης( recall ) και χρησιμοποιείται ως στατιστικό μέτρο για την αξιολόγηση της απόδοσης.

### Αποτελέσματα πρώτης εκτέλεσης

**Sequential :**

1136/1136 [==============================] - 9s 8ms/step - loss: 1.6665 - accuracy: 0.5318

Loss: 1.666527509689331

Accuracy: 0.5317971706390381

**Xgboost :**

precision recall f1-score support

accuracy 0.77 36324

macro avg 0.79 0.53 0.60 36324

weighted avg 0.77 0.77 0.74 36324

**RandomForestClassifier:**

precision recall f1-score support

accuracy 0.78 36324

macro avg 0.86 0.55 0.63 36324

weighted avg 0.83 0.78 0.76 36324

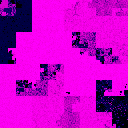
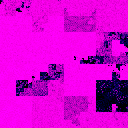
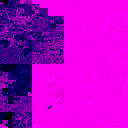
Το μοντέλο Sequential παρουσίασε ποσοστό ακριβείας 53% ενώ προηγουμένως, το malimg dataset εμφάνισε 95%. Βασικό πρόβλημα του καινούριου dataset φαίνεται να ήταν το μέγεθος του και ο μεγάλος αριθμός εικόνων ορισμένων οικογενειών, δηλαδή οι εικόνες δεν ήταν κατανεμημένες ισορροπημένα για κάθε κλάσση. Παρ’όλα αυτά, τα άλλα 2 μοντέλα εμφάνισαν ποσοστό ακρίβειας 77 και 78 τοις εκατό. Στη συνέχεια, τέθηκε το ερώτημα εάν ο τρόπος μετατροπής των λογισμικών σε εικόνες .png δεν ήταν αποδοτικός και έτσι αναζητήθηκε ένας καινούριος.

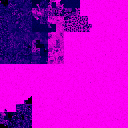
## **Δεύτερη μετατροπή**

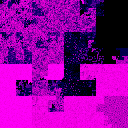
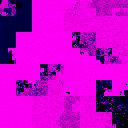
Σε επικοινωνία με τον καθηγητή μου , μελετήθηκε η βιβλιοθήκη της Python scurve και με τον κώδικα στο αρχείο binvis έγινε η νέα μετατροπή των εκτελέσιμων αρχείων σε εικόνες .png . Πιο συγκερκριμένα, έγινε αναζήτηση στους χάρτες χρωμάτων ["class", "hilbert", "entropy", "gradient"], στο μέγεθος των σχηματισμένων εικόνων και στην αναλογία διαστάσεων εικόνας με σκοπό να διασφαλιστεί η καλύτερη ακρίβεια. Η μεταβλητή pixel layout map αποτέλεσε τον χάρτη διάταξης των pixel της κάθε δημιουργημένης εικόνας στον οποίο επιλέχθηκε η τιμή hilbert( διέλευση όλων των σημείων από την καμπύλη πλήρωσης χώρου hilbert ). Ο χάρτης χρωμάτων με την μεγαλύτερη ακρίβεια ήταν αυτός με την τιμή entropy ο οποίος χρησιμοποιήθηκε. Τέλος, αποφασίστηκε οι εικόνες να είναι τετραγωνισμένων διαστάσεων ( squared ) και το μέγεθος τους 64x64 pixel για τη σύγκριση των αποτελεσμάτων με την πρώτη μετατροπή. Ο αριθμός των αρχείων επέκτασης .png και των οικογενειών για την 2η εκτέλεση ήταν ακριβώς ο ίδιος με αυτόν της 1ης και μετά τη δεύτερη μετατροπή.

### Παράδειγμα δεύτερης μετατροπής

**Τα 9 ίδια δείγματα του malware AveMariaRAT**







### Χαρακτηριστικά εκτέλεσης

**Τα χαρακτηριστικά της εκτέλεσης ήταν τα ίδια με αυτά της 1ης , με σκοπό να αναγνωριστούν οι διαφορές των αποτελεσμάτων.**

### Αποτελέσματα δεύτερης εκτέλεσης

**Sequential :**

1136/1136 [==============================] - 9s 8ms/step - loss: 1.4075 - accuracy: 0.5739

Loss: 1.4074965715408325

Accuracy: 0.5739180445671082

**Xgboost :**

precision recall f1-score support

accuracy 0.78 36324

macro avg 0.80 0.54 0.62 36324

weighted avg 0.78 0.78 0.76 36324

**RandomForestClassifier:**

precision recall f1-score support

accuracy 0.78 36324

macro avg 0.87 0.53 0.61 36324

weighted avg 0.83 0.78 0.75 36324

Μετά από την δεύτερη εκτέλεση, έγινε αντιληπτό ότι για το ίδιο σύνολο δεδομένων,τον ίδιο αριθμό εικόνων και τις ίδιες παραμέτρους των μοντέλων που χρησιμοποιήθηκαν, τα αποτελέσματα για τα 2 μοντέλα( xgboost, RandomForestClassifier ) ήταν σχεδόν ίδια, ενώ για το Sequential παρατηρήθηκε μία πολυ μικρή βελτίωση ακρίβειας της τάξης 4%. Αυτή τη στιγμή ακολούθησαν σκέψεις και συμπεράσματα για την συνολική εικόνα της εργασίας .

# 

# **Σύνοψη**

Τα βήματα που ακολουθήθηκαν για την εκπόνηση της εργασίας ήταν τα εξής:

* Παρατηρήθηκε το πρόβλημα της κακόβουλης χρήσης των λογισμικών σήμερα και των επιπτώσεων της χρήσης αυτής και αποφασίστηκε η ενασχόληση με αυτόν τον τομέα επικοινωνώντας με τον επιβλέπων καθηγητή μου.
* Ύστερα απο μελέτη του malimg dataset και του σχετικού άρθρου για την ταξινόμηση εικόνων, μετά από τη λήψη των αρχείων και τη δημιουργία ενός συνελικτικού νευρωνικού δικτύου ακολούθησε η εκτέλεση του και τα αποτελέσματα. Συνολικά, έπειτα από την ενασχόληση με το malimg dataset έγινε αντιληπτός ο τρόπος λειτουργίας σε πιο προχωρημένα και πιο σύνθετα στάδια.
* Έγινε λήψη των εκτελέσιμων αρχείων επέκτασης .exe από το csv που παρείχε το Maware Bazaar και μετατροπή αυτών σε ασπρόμαυρες εικόνες με την πρώτη τεχνική. Ακολούθησαν, η προσθήκη νέων μοντέλων-αλγορίθμων, διαδικασιών και βιβλιοθηκών στον κώδικα, μελέτη των παραμέτρων τους και προσπάθεια για την εξασφάλιση της καλύτερης δυνατής ακρίβειας. Μετά την πρώτη εκτέλεση, φάνηκε οτι τα αποτελέσματα της δεν ήταν σαφή, με δυο εκ των τριών μοντέλων( Xgboost, RandomForestClassifier ) να έχουν καλό ποσοστό ακρίβειας(77,78 τοις εκατό) ενώ το άλλο ( Sequential ) μέτριο(53 τοις εκατό). Μετά τα πρώτα αποτελέσματα αναζητήθηκε μία καλύτερη μέθοδος μετατροπής των αρχείων σε εικόνες για να διαπιστωθεί εάν και που υπάρχει το πρόβλημα ταξινόμησης.
* Η εργασία κλείνει μετατρέποντας τα αρχεία σε εικόνες με τη βοήθεια του κώδικα στο αρχείο binvis της βιβλιοθήκης scurve της python, το οποίο λειτουργεί με μια καινούρια προοπτική απεικόνισης των δυαδικών αρχείων μέσω της μετατροπής τους χάρη στις καμπύλες πλήρωσης χώρου ( space-filling curves ). Τα καινούρια αποτελέσματα όμως μετά την δεύτερη εκτέλεση είναι και πάλι πολύ κοντά με αυτά της πρώτης εκτέλεσης.

Συμπερασματικά, διαπιστώθηκε :

* Το δύνολο δεδομένων που χρησιμοποιήθηκε δεν ήταν κατανεμημένο/ισορροπημένο. Παρ’όλο που δώθηκαν τα βάρη που αντιστοιχούσαν στην κάθε κλάσση, υπήρχαν κλάσσεις με πολύ μεγάλο η πολύ μικρό αριθμό εικόνων με αποτέλεσμα να μειώνεται η ακρίβεια του δικύου.
* Κάποιες εικόνες ήταν πολύ χαρακτηριστικές σχετικά με την κλάσση τους, με αποτέλεσμα να ταξινομούνται σωστά ενώ κάποιες άλλες το αντίθετο.
* Εξαιτίας του μεγάλου χώρου μνήμης RAM που δέσμευαν οι εικόνες , δεν λήφθηκαν τα πιο σαφή αποτελέσματα αφού ήταν μόνο μεγέθους 64x64 pixels.
* Τα αποτελέσματα θα ήταν καλύτερα έαν είχε γίνει η διαδικασία hyperparameter tuning (αναζήτηση των καλύτερων παραμέτρωνγια τα μοντέλα-τους αλγοριθμους), γεγονός που ήθελε πολύ χρόνο.

# 

# **Βιβλιογραφία**

<https://en.wikipedia.org/wiki/Main_Page>

<https://github.com/cortesi/scurve/blob/master/binvis>

<https://towardsdatascience.com/malware-classification-using-convolutional-neural-networks-step-by-step-tutorial-a3e8d97122f>

<https://stackoverflow.com/>

<https://xgboost.readthedocs.io/en/stable/python/python_intro.html>

<https://scikit-learn.org/stable/>

<https://keras.io/>

<https://www.python.org/>

<https://numpy.org/>

<https://matplotlib.org/>

<https://www.anaconda.com/products/distribution>

<https://www.kali.org/>

<https://www.statista.com/statistics/680953/global-malware-volume/>

<https://www.comparitech.com/antivirus/malware-statistics-facts/>

<https://dataprot.net/statistics/malware-statistics/>