Universitatea Națională de Știință și Tehnologie POLITEHNICA București

Facultatea de Automatică și Calculatoare

Detectarea tehnicilor de steganografie si a malware-ului din imagini

Student: Vădana Ioan-Grigore

Profesor Indrumator: Mocanu Stefan Alexandru

2024

Cuprins

Introducere

Proiectul dezvoltat este destinat analizei și detectării tehnicilor de steganografie și a malware-ului din imagini.

În contextul actual al securității cibernetice, protecția datelor și prevenirea accesului neautorizat la informații sunt imperative. Una dintre tehnicile folosite pentru protejarea și ascunderea informațiilor este steganografia, care permite încorporarea datelor secrete în fișiere media, cum ar fi imagini, audio sau video, fără a atrage atenția. Această tehnică nu doar că ascunde informațiile, dar face și detectarea acestora extrem de dificilă pentru persoanele neautorizate.

Malware-ul este adesea mascat în fișiere aparent inofensive, complicând astfel eforturile de detectare. Acest proiect abordează problema detectării malware-ului printr-o metodă inovatoare care convertește fișierele Portable Executable (PE) în imagini de tip byteplot. Această conversie transformă problema analizei fișierelor executabile, de obicei bazată pe semnătură sau pe comportament, într-una care poate fi abordată cu tehnici vizuale utilizând învățarea automată. Aplicând modele de învățare profundă pe aceste imagini, proiectul își propune să identifice și să clasifice diverse tipuri de malware într-o manieră mai intuitivă și vizuală.

Aplicația are o interfață bazată pe tab-uri, fiecare tab reprezentând un tip diferit de analiză. Utilizatorii pot naviga ușor între diferitele metode de analiză, iar rezultatele sunt afișate clar în panouri de text. Rolul acestui proiect este de a oferi unelte de securitate cibernetică care pot detecta ascunderea informațiilor (steganografia) și prezența malware-ului în fișierele de imagine

- Obiectivele principale:

<u>Detectarea steganografiei din imagini:</u>

Proiectul include crearea unei aplicații pentru detectarea steganografiei folosind tehnici precum analiza LSB, analiza histogramelor, transformările Fourier și Wavelet, și analiza texturilor. Această aplicație va permite identificarea și raportarea cazurilor de steganografie în imagini.

1. **Detectarea Modificărilor LSB (Least Significant Bit)**: Identificarea schimbărilor subtile la nivelul biților puțin semnificativi ai pixelilor, o metodă comună în steganografie.
2. **Analiza Histogramelor**: Examinarea distribuției valorilor de intensitate ale pixelilor pentru a detecta posibile inserții de informații.
3. **Utilizarea Transformatei Fourier**: Explorarea componentelor frecvențiale ale imaginilor pentru a identifica anomalii care pot indica prezența steganografiei.
4. **Aplicarea Transformării Wavelet**: Detectarea schimbărilor la diferite niveluri de rezoluție ale imaginii, utilă pentru identificarea inserțiilor complexe.
5. **Analiza Texturii**: Evaluarea modificărilor texturii utilizând matricea de co-ocurență a nivelelor de gri pentru a identifica steganografia avansată.

Detectarea malware-ului din imagini:

Scopul acestei implementari este de a prezenta în detaliu metodologia propusă, începând cu conversia datelor binare ale fișierelor în imagini și până la aplicarea și evaluarea modelelor de învățare profundă pentru clasificarea acestor imagini. Obiectivele specifice includ:

1. **Detalierea procesului de conversie** a fișierelor PE în imagini de tip byteplot, explicând metodologia și tehnicile folosite pentru transformarea datelor binare în format vizual.
2. **Explorarea seturilor de date** Malimg și Malevis, care sunt frecvent utilizate în acest tip de clasificare a imaginilor. Se va discuta despre caracteristicile, avantajele și limitările acestor seturi de date în contextul detectării malware-ului.
3. **Dezvoltarea și optimizarea unui model de clasificare**, utilizând arhitecturi de rețele neuronale convoluționale (CNN) adaptate pentru a lucra eficient cu imaginile generate din fișierele PE.
4. **Evaluarea performanței modelului** prin metrici standardizate precum acuratețea și pierderea în cadrul unui cadru de validare riguros.

Prezentare suport tehnic

În domeniul detectării steganografiei, au fost dezvoltate diverse metode și aplicații care folosesc tehnici avansate pentru a identifica și extrage datele ascunse. De exemplu, Ker et al. (2005) au discutat utilizarea analizei statistice pentru detectarea steganografiei în imagini digitale, evidențiind că modificările minore ale histogramelor pot fi detectate eficient cu metode statistice avansate.

Un alt exemplu relevant este lucrarea lui Fridrich și Goljan (2004), în care au fost explorate transformările wavelet pentru a descoperi steganografia, demonstrând că aceste transformări pot evidenția anomalii la niveluri diferite de rezoluție care ar putea fi trecute cu vederea în analizele spațiale normale.

Aplicarea transformării Fourier pentru identificarea steganografiei a fost detaliată în lucrările lui Provos și Honeyman (2003), care au arătat că analiza frecvențelor poate dezvălui informații ascunse în componentele de frecvență ale unei imagini.

Proiectul utilizează tehnologii de vârf în domeniul procesării de imagini și învățării automate, inclusiv TensorFlow și Keras, care permit implementarea eficientă a modelelor complexe de rețele neuronale convoluționale. Arhitectura MobileNetV2 a fost aleasă pentru eficiența sa în termeni de cost computațional și performanță, fiind ideală pentru aplicații unde resursele de calcul sunt o considerație critică.

Studiile anterioare au demonstrat eficacitatea abordărilor vizuale în detectarea malware-ului, cum ar fi lucrările lui **Nataraj et al. [1]**, care au utilizat histograma de orientare a gradientului pentru a clasifica imaginile malware, și **Makandar și Patrot [2]**, care au aplicat învățarea profundă

pe imagini similare pentru a distinge între diferite familii de malware. Aceste studii validează abordarea vizuală ca fiind promițătoare pentru detectarea avansată a malware-ului.

Prezentarea tehnica

Componentele principale ale aplicației

1. **Interfața Grafică (GUI)**: Construită folosind tkinter și ttk pentru a oferi o interfață intuitivă și accesibilă. Utilizatorii pot naviga între diferite tab-uri pentru a încărca imagini și a executa diverse tipuri de analize. Exemplu de utilizare:

1) 

2. **Panouri de Analiză**:
    - **Încărcarea Imaginii**: Permite utilizatorilor să încarce o imagine de pe dispozitivul lor pentru a fi analizată.
    - **Analiza LSB (Least Significant Bit)**: Detectează modificările la nivelul celor mai puțin semnificativi biți ai pixelilor.
    - **Analiza Histogramă**: Examinează distribuția valorilor de intensitate ale pixelilor.
    - **Analiza Fourier**: Utilizează transformata Fourier pentru a identifica frecvențele componentelor.
    - **Analiza Wavelet**: Aplică transformarea wavelet pentru a examina detalii la diferite niveluri de rezoluție.

- **Analiza Texturii**: Folosește matricea de co-ocurență pentru a evalua textura imaginii.
- **Analiza Malware**: Detectează potențialele amenințări malware prin clasificarea imaginilor.

Biblioteci și Tehnologii Utilizate

- **Tkinter și Ttk**: Utilizate pentru construcția GUI.
- **Pillow (PIL Fork)**: Manipulare și procesare imagini.
- **NumPy**: Procesare și analiză numerică a datelor.
- **Matplotlib**: Vizualizare grafică a datelor.
- **Scipy**: Algoritmi de analiză a datelor, cum ar fi **find_peaks** pentru identificarea vârfurilor în histogramă.
- **PyWavelets**: Implementarea transformării wavelet.
- **Skimage**: Analiza texturilor prin **graycomatrix** și **graycoprops**.
- **TensorFlow Keras**: Învățare automată pentru analiza malware.

Algoritmi și Funcționalități

- **Analiza LSB**: Folosesc un test chi-pătrat pentru a detecta distribuții neuniforme ale biților LSB, ceea ce poate indica steganografia. Exemplu:

Analiza LSB:

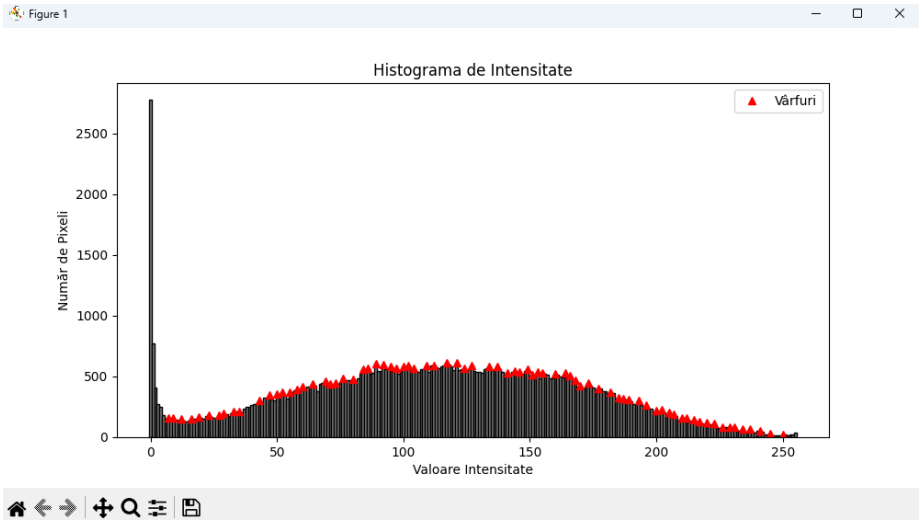
- Canalul Roșu: 9503 '0', 8497 '1' (chi2=56.22, p=0.000)

- Canalul Verde: 9390 '0', 8610 '1' (chi2=33.80, p=0.000)

- Canalul Albastru: 9593 '0', 8407 '1' (chi2=78.14, p=0.000)

Steganografie detectată

- **Analiza Histogramă**: Identifică vârfurile neobișnuite în distribuția pixelilor și calculează caracteristici statistice cum ar fi asimetria și curtosis.



2)

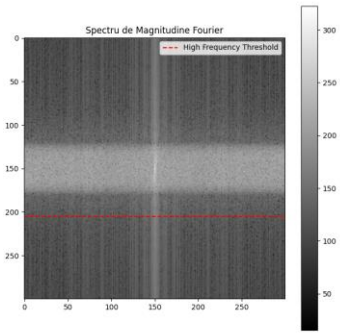
Vârfuri identificate la intensități: 7, 9, 12, 16, 19, 23, 27, 29, 33, 35, 43, 47, 50, 52, 55, 58, 60, 64, 69, 71, 73, 76, 80, 84, 86, 89, 92, 95, 97, 100, 102, 104, 109, 112, 117, 121, 124, 127, 134, 137, 141, 144, 146, 149, 151, 153, 155, 160, 164, 166, 168, 170, 173, 177, 182, 185, 187, 189, 193, 196, 200, 202, 205, 207, 210, 212, 215, 217, 220, 223, 226, 229, 231, 234, 237, 241, 245, 250

Skewness: 3.789 (Indică o distribuție nesimetrică)

Kurtosis: 37.205 (Indică o prezență a vârfurilor accentuate)

Concluzie: Analiza sugerează indicii puternice de steganografie sau manipulare a imaginii.

- **Analiza Fourier**: Transformata Fourier dezvăluie frecvențe neobișnuite sau manipulări în spectrul de frecvență al imaginii.



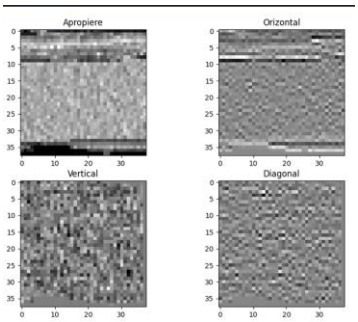
3)

Analiza Fourier: Fourier - Media: 134.87, STD: 35.04, Skew: 1.01, Kurtosis: 0.28

Zonele cu frecvențe înalte indică potențiale manipulări sau informații ascunse.

Concluzie: Nu există indicii semnificative de manipulare sau steganografie.

- **Analiza Wavelet**: Descompune imaginea în coeficienți de apropiere și detalii, analizând energia acestora pentru indicii de steganografie.

4) 

Analiza Wavelet: Energie Orizontală: 11842397.42, Energie Verticală: 17717312.48, Energie Diagonală: 3705720.67

Energia mare în componentele orizontale și diagonale poate indica manipulare sau inserție de date.

Concluzie: Nu există indicii semnificative de manipulare sau steganografie.

- **Analiza Texturii**: Evaluează proprietăți ale texturii precum contrastul și omogenitatea, care pot varia semnificativ în prezența steganografiei.

Analiza Texturii:

Contrast: 3208.4255

Dissimilarity: 40.3924

Asm: 0.0005

Energy: 0.0221

Homogeneity: 0.0665

Concluzie: Detaliile texturale pot indica zone cu manipulări potențiale sau caracteristici neobișnuite ale texturii.

- **Analiza Malware**: Folosesc un model pre-antrenat pentru a clasifica imaginea în diverse categorii de malware, ajutând la detectarea amenințărilor ascunse.

  Rezultatele clasificării Malware:

  Clasa HackKMS cu probabilitatea 100.00%

  Clasa Allaple cu probabilitatea 0.00%

  Clasa VBKyrpy cu probabilitatea 0.00%

  Clasa Regrun cu probabilitatea 0.00%

  Clasa Injector cu probabilitatea 0.00%

API: MobileNetV2 din Keras Applications este folosit ca model de bază datorită eficienței sale în calculul caracteristicilor din imagini și adaptabilității la noi probleme prin transferul de învățare.

Algoritm: Acest model preantrenat pe ImageNet este utilizat pentru a extrage trăsături bogate din imagini, ceea ce ajută în clasificarea precisă a datelor.

Metodologie

Structura Directorilor:

Directoarele de date, train_dir și val_dir, sunt configurate pentru a conține imaginile de antrenament și validare, respectiv. Acestea sunt esențiale pentru încărcarea și prelucrarea seturilor de date în model.

Generatoare de Date:

- **Antrenament**: Utilizează **ImageDataGenerator** pentru augmentarea datelor, inclusiv scalare, rotație, translație și flipare orizontală, care ajută la generalizarea modelului și prevenirea overfitting-ului.

5)
```
# Generator de date pentru antrenament cu redimensionare și augmentare minimă
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True
)
```

- **Validare**: Folosesc un generator simplu care doar reeșalonează pixelii imaginilor, asigurând că inputul modelului este normalizat corespunzător.

Fluxul de Date:

- Datele sunt încărcate direct din directoarele specificate și sunt prelucrate în batch-uri de 32, fiind ajustate la dimensiunea de 224x224 pixeli necesară pentru inputul în MobileNetV2.

Arhitectura Modelului:

- **MobileNetV2**: Ales pentru eficiența sa în procesarea imaginilor. Modelul este încărcat cu greutăți pre-antrenate de pe ImageNet și fără straturile superioare, adaptându-l pentru clasificarea malware-ului.
- **Personalizări ale Modelului**: Include GlobalAveragePooling2D, un strat dens de 1024 de neuroni cu activare 'relu', normalizare batch, și dropout pentru regularizare. Ultimul strat este un strat dens cu activare 'softmax' pentru clasificarea în 31 de clase diferite.

6)
```python
# Utilizarea MobileNetV2 ca model de bază, înghețarea straturilor preantrenate
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model.trainable = False

# Adăugarea straturilor superioare
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dense( units: 1024, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense( units: 31, activation='softmax')
])
```

Compilarea Modelului:

- Folosește optimizatorul Adam cu o rată inițială de învățare de 0.001 și funcția de pierdere **categorical_crossentropy**, ideală pentru problemele de clasificare multi-clasă.

7)
```python
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

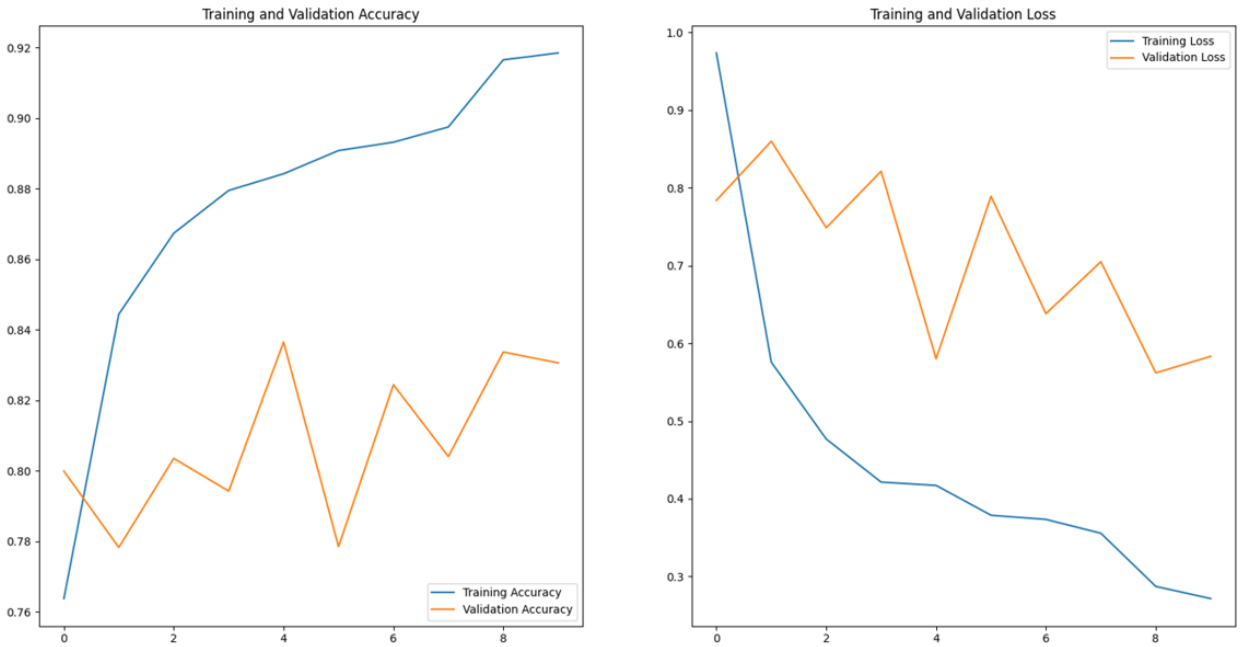
Antrenarea și Evaluarea Modelului

Procesul de Antrenament:

- Modelul este antrenat pentru 10 epoci, utilizând datele de antrenament și validare. Performanța este monitorizată prin acuratețe și pierdere pentru a evalua progresul modelului.

Evaluare Finală:

- Modelul este evaluat pe setul de validare pentru a obține pierderea finală și acuratețea, furnizând metrici finale care confirmă calitatea modelului.

8)


Graficul din Stânga: Acuratețea de Antrenament și Validare

- **Axa X**: Numărul de epoci de antrenament.
- **Axa Y**: Procentul de acuratețe, unde 1.0 reprezintă 100% acuratețe.
- **Linia Albastră** reprezintă acuratețea modelului pe setul de antrenament.
- **Linia Portocalie** reprezintă acuratețea modelului pe setul de validare.

Observații:

- Acuratețea de antrenament (**linia albastră**) crește constant pe măsură ce modelul învață din date, indicând o îmbunătățire a modelului în identificarea corectă a claselor din setul de antrenament.
- Acuratețea de validare (**linia portocalie**) prezintă variații mai mari și nu crește la fel de constant ca acuratețea de antrenament, sugerând că modelul poate avea dificultăți în generalizarea pe date noi nevăzute în timpul antrenamentului.

Graficul din Dreapta: Pierderea de Antrenament și Validare

- **Axa X**: Numărul de epoci de antrenament.
- **Axa Y**: Valoarea funcției de pierdere, unde valori mai mici indică o performanță mai bună.
- **Linia Albastră** indică pierderea pe setul de antrenament.
- **Linia Portocalie** indică pierderea pe setul de validare.

Salvarea și Utilizarea Modelului

Salvarea Modelului:

- Modelul este salvat în formatul '.keras', asigurându-se că poate fi încărcat și utilizat ulterior pentru predicții sau pentru continuarea antrenamentului.

9)
```
# Salvarea modelului în formatul recomandat
model.save('high_accuracy_malware_model.keras')
```

Contribuția Personală

Contribuția mea a inclus:

- **Integrarea și adaptarea algoritmilor**: Selectarea și implementarea algoritmilor specifici fiecărui tip de analiză, asigurând acuratețea și eficiența lor.
- **Optimizarea interfeței**: Crearea unei interfețe prietenoase și eficiente, care îmbunătățește experiența utilizatorului final.

- Am personalizat arhitectura MobileNetV2 prin adăugarea straturilor specifice care să servească nevoilor de clasificare a malware-ului din imagini. Acest lucru a inclus configurarea dimensiunilor straturilor, a funcțiilor de activare și a algoritmului de regularizare pentru a maximiza performanța specifică datelor de malware.
- Am implementat modelul folosind Keras, ajustând hiperparametrii și evaluând diferite configurații ale modelului pentru a obține cele mai bune performanțe. Am folosit, de asemenea, tehnici de validare încrucișată pentru a asigura robustețea modelului.

Prezentare mod de utilizare

Aplicația este concepută pentru a fi intuitivă și ușor de utilizat, oferind o serie de funcționalități accesibile printr-o interfață grafică clară. Aici sunt detaliați pașii principali și interacțiunea cu utilizatorul, împreună cu configurările disponibile pentru a personaliza experiența de utilizare.

Lansarea Aplicației

Utilizatorii pot deschide aplicația făcând dublu clic pe executabilul programului. La deschidere, se va afișa fereastra principală a aplicației, care include mai multe tab-uri, fiecare dedicat unei anumite analize. Interfața este simplă, cu controale clare și butoane de navigare între diferitele funcții.

Încărcarea și Selecția Imaginilor

Executarea Analizelor

2. Selectarea Tipului de Analiză: După încărcarea imaginii, utilizatorii pot naviga între tab-uri pentru a selecta tipul de analiză pe care doresc să îl efectueze: Analiza LSB, Analiza Histogramă, Analiza Fourier, Analiza Wavelet, Analiza Texturii și Analiza Malware.
3. Executarea Analizei: Fiecare tab conține un buton (ex. „Analizează LSB”), care, odată apăsat, inițiază procesul de analiză. Procesarea este automată, și rezultatele vor fi afișate în zona de text sau grafic a tab-ului respectiv.

Interpretarea Rezultatelor

4. Vizualizarea și Interpretarea Rezultatelor: Rezultatele fiecărei analize sunt prezentate direct în tab-ul corespunzător sub formă de text explicativ sau diagrame. Utilizatorii pot vedea detalii despre posibilele anomalii detectate sau caracteristicile imaginii analizate.

Configurare și Personalizare

5. Configurări: Utilizatorii pot accesa setările aplicației printr-un meniu de configurare, unde pot ajusta opțiunile precum calea implicită pentru salvarea rezultatelor, nivelul de detaliu al rapoartelor sau preferințele de vizualizare.

6. Ajutor și Suport: Un tab sau meniu de ajutor oferă instrucțiuni suplimentare și resurse pentru utilizatori, ajutându-i să înțeleagă mai bine funcționalitățile și să rezolve problemele întâmpinate.

Finalizarea Utilizării

7. Închiderea Aplicației: Utilizatorii pot închide aplicația oricând, folosind butonul de închidere standard din colțul ferestrei sau printr-o opțiune de „Închidere” din meniul principal.

Concluzii:

Îndeplinirea Obiectivelor pentru analiza steganografiei:

1. **Analiza LSB (Least Significant Bit)**: A fost implementată cu succes, oferind utilizatorilor capacitatea de a detecta modificări subtile în LSB care pot indica prezența steganografiei.
2. **Analiza Histogramă**: Aplicația permite analiza detaliată a histogramelor pentru a identifica anomalii care ar putea fi semne ale inserției de date.
3. **Analiza Fourier**: Transformata Fourier este utilizată eficient pentru a dezvălui manipulări ale frecvențelor sau alte semne ale steganografiei.
4. **Analiza Wavelet**: Implementarea transformării wavelet ajută la observarea detaliilor care sunt altfel greu de detectat, contribuind la o analiză mult mai profundă.
5. **Analiza Texturii**: Matricea de co-ocurență a nivelurilor de gri furnizează o perspectivă detaliată asupra texturii, oferind indicii valoroase despre posibila prezență a datelor ascunse.

Prin combinarea tehnologiilor avansate cu o interfață accesibilă, "Detector de Steganografie Avansat" nu doar că atinge obiectivele propuse, dar setează și un standard în detectarea steganografiei, demonstrând cum tehnologia poate fi folosită pentru a îmbunătăți securitatea digitală și educația în domeniul securității cibernetice. Această aplicație este o dovadă a impactului pozitiv pe care tehnologia aplicată o poate avea în lupta împotriva amenințărilor informatice.

Îndeplinirea Obiectivelor pentru detectarea malware-ului:

1. Conversia fișierelor PE în imagini a fost realizată eficient, permitând o analiză vizuală detaliată și transformând un set de date tradițional într-un format compatibil cu tehnicile de învățare profundă.
2. Dezvoltarea modelului de clasificare bazat pe MobileNetV2 a arătat o performanță remarcabilă, integrând cu succes tehnici avansate de prelucrare a imaginilor și învățare automată pentru a identifica caracteristicile relevante ale malware-ului.

3. Evaluarea și validarea modelului au confirmat eficacitatea acestuia, cu rezultate de acuratețe și pierdere care subliniază capacitatea modelului de a funcționa bine în scenarii reale, oferind o bază solidă pentru utilizarea în sisteme de securitate.

mpactul acestui proiect este vast, având potențialul de a transforma modul în care organizațiile abordează securitatea informațiilor:

Îmbunătățirea securității cibernetice: Integrarea acestui model în sistemele de securitate existente poate crește semnificativ capacitatea de apărare a rețelelor, protejând datele și infrastructura critică.

Scalabilitate și adaptabilitate: Modelul poate fi adaptat și scalat pentru a se potrivi cu diferite medii și cerințe, făcându-l ideal pentru utilizare în diverse scenarii industriale

Referinte bibliografie

[1]:

[*Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011). Malware images: visualization and automatic classification. Proceedings of the 8th International Symposium on Visualization for Cyber Security*].

[2]:

[*Makandar, A., & Patrot, A. (2015). Deep learning approach for malware classification. 2nd International Conference on Electronics and Communication Systems (ICECS)*].

---

https://www.researchgate.net/publication/41099668_Steganalysis_algorithms_for_detecting_the_hidden_information_in_image_audio_and_video_cover_media

https://www.mdpi.com/2076-3417/13/21/11771#:~:text=The%20LSB%20approach%20replaces%20the,sufficient%20to%20hide%20the%20message.

https://www.researchgate.net/publication/309185486_Study_of_Image_steganography_using_LSB_DFT_and_DWT

https://dl.acm.org/doi/fullHtml/10.1145/3503047.3503081

https://www.geeksforgeeks.org/image-based-steganography-using-python/

https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/

https://www.geeksforgeeks.org/adam-optimizer/

https://paperswithcode.com/method/mobilenetv2