**Alegerea și Motivarea Limbajelor de Programare**

Pentru dezvoltarea sistemului avansat de detectare a phishing-ului, am efectuat o analiză atentă a limbajelor de programare și a framework-urilor disponibile, luând în considerare aspecte precum performanța, flexibilitatea și suportul comunității. Am optat pentru o combinație strategică între Python și JavaScript, împreună cu framework-ul React.

**Python**

**Motivare**: Python oferă un echilibru perfect între simplitate și putere, facilitând dezvoltarea rapidă și eficientă a componentelor server-side ale sistemului. Ecosistemul său bogat, cu biblioteci precum TensorFlow pentru învățarea automată și Flask pentru dezvoltarea web, ne-a oferit instrumentele necesare pentru a crea o aplicație solidă și ușor de întreținut.

**Alternative Analizate:**

* **Java**: Considerat pentru performanța sa, dar a fost respins din cauza complexității excesive și a timpului de dezvoltare mai lung.

**JavaScript** - Interactivitatea și Rezponsivitatea în Interfața Utilizator

**Motivare**: JavaScript este esențial pentru a crea o interfață utilizator reactivă și dinamică. Împreună cu framework-ul React, oferă un model de dezvoltare componentizat, accelerând dezvoltarea front-end și asigurând o experiență fluidă pentru utilizatori.

**Alternative Analizate:**

* **Angular**: A fost evaluat pentru structura sa rigidă, dar React a fost preferat pentru flexibilitatea sa și curba de învățare mai redusă.
* **Vue.js**: O alternativă viabilă, dar React a fost preferat datorită popularității sale crescânde și suportului extins.

**React**

**Motivare**: React utilizează o reprezentare virtuală a DOM-ului pentru a minimiza actualizările reale ale interfeței utilizatorului. Această abordare aduce beneficii semnificative în ceea ce privește performanța și eficiența aplicației, în special în aplicații cu interfețe complexe.

**Modelarea Sistemului de Detectare a Phishing**

**3.1 Introducere**

În acest capitol, se va prezenta în esența modelării sistemului de detecție a phishing-ului prin email, dezvăluind aspectele tehnice ce stau la baza arhitecturii și algoritmilor ce compun acest sistem de detecție.

**3.2 Analiza Conținutului Email-urilor Phishing: Explorarea Mecanismelor de Detectare Avansate**

**3.2.1 Filtrarea Dinamică a Link-urilor**

Tehnologie de Evaluare în Timp Real: Se utilizează un sistem dinamic de evaluare a link-urilor, care analizează parametri precum redirecționările, autenticitatea DNS-ului și semnăturile cunoscute ale site-urilor phishing.

**3.2.2 Analiza Semantică a Conținutului**

Procesare Avansată a Limbajului Natural: Se implementează algoritmi avansați de procesare a limbajului natural (NLP) pentru a analiza cuvinte-cheie, structuri gramaticale și contextul semantic al conținutului email-urilor. Această analiză inteligentă permite detecția schemelor de manipulare în textul mesajelor.

**3.3 Colectarea Datelor**

**3.3.1 Setul de Date de Antrenament**

Detalierea procesului de colectare a datelor de antrenament, inclusiv sursele de date, criteriile de selecție și preprocesarea acestora.

**3.3.2 Setul de Date de Testare**

Descrierea setului de date utilizat pentru evaluarea performanței sistemului, inclusiv criteriile de selecție și metodele de preprocesare.

**3.4 Extracția de Caracteristici**

3.4.1 Tehnici de Extracție

Prezentarea algoritmilor și tehnicilor folosite pentru a extrage caracteristicile semnificative din datele brut ale email-urilor.

**3.5 Algoritmi de Clasificare**

3.5.1 Algoritmul de Învățare Automată

3.5.2 Antrenarea Modelului

Detalierea procesului de antrenare a modelului și evaluarea performanței pe setul de date de antrenament.

**3.6 Decizia de Detectare**

3.6.1 Criterii de Decizie

Explicarea criteriilor pe baza cărora sistemul decide dacă un email este sau nu phishing.