Analiza Comparativă a Tehnicilor de Învățare Automată pentru Detecția Atacurilor de Tip Phishing

Stoie Georgiana-Ștefania

March 2025

1 Introducere

Internetul a devenit o parte indispensabilă din viața noastră, însă, s-au oferit o mulțime de oportunități pentru a realiza activități malițioase anonime de tip Phishing. Cu toate că au fost dezvoltate metode pentru a reunoaște aceste activități malițioase, Phisherii au reușit să-și evolueze metodele pentru a evita aceste metode. Unele dintre cele mai bune metode care a avut succes în detectarea acestor activități este învățarea automată. Acest proiect se bazează pe analiza comparativă a unor tehnici de învățare automată pentru a detecta atacurile de tip Phishing [10].

2 Context

Există numeroare tipuri de phishing, incluzând: vishing, spear phishing, whaling și email phishing. În 1990, phishing a fost raportat pentru prima dată, fiind folosit pentru a fura parole. Însă, în ultimii ani, atacurile de tip phishing au înregistrat o creștere semnificativă. URL phishing este unul dintre atacuri.

Un URL este o adresă website care semnifică locația unui webite pe o rețea. URL-urile sunt împărțite în două categorii: malițioase și cele inofensive. Detectarea URL-urilor malițioase necesită extragerea unor caracteristici, fiind comparate ulterior pentru a determina dacă acel URL este malicios sau inofensiv [6].

3 Metodologie

3.1 Algoritmi de Învățare Automată

1. Long-Short Term Memory (LSTM)
2. Random Forest

3. K-Nearest-Neighbours (KNN)

3.2 Data Set

În acest set de date s-au folosit 10000 de URL-uri, dintre care 5000 URL-uri malițioase iar restul de 5000 URL-uri inofensive [12].

3.3 Caracteristicile Extrase

Caracteristicile extrase sunt în număr de 15, fiecare luate în considerare.

1. Prezența IP-ului
2. Prezența simbolului @
3. Lungime
4. Adâncime
5. Pozitia "//" în URL
6. HTTP/HTTPS în Domeniu
7. Metode de scurtare a URL-ului
8. Prezența ' - ' în URL
9. Trafic Web
10. Vârsta Domeniului
11. Sfârșitul Domeniului
12. Prezența IFrame
13. Mouse Over
14. Right Click
15. Numărul de Forwarding-uri

4 Rezultate

În cadrul acestui proiect, au fost realizate teste pe diverse procente ale seturilor de date destinate antrenării și testării. În continuare se vor prezenta procentele, în forma: date antrenare-testare: 90-10, 80-20, 70-30, 60-40, 50-50, 40-60, 30-70, 20-80.

Rezultatele nu au prezentat variații semnificative, astfel, scăderea procentajelor nu a avut un impact semnificativ asupra performanței modelului.

4.1 LSTM

În figura 1, este prezentată acuratețea obținută la testare pentru toate seturile de date corespunzătoare procentelor menționate anterior.

LSTM		
Train - Test Size	Accuracy	
	90-10	1.0
	80-20	1.0
	70-30	1.0
	60-40	1.0
	50-50	1.0
	40-60	1.0
	30-70	1.0
	20-80	1.0
	10-90	0.997

Figure 1: Acuratețea modelului în funcție de diferite procente alocate

După cum se poate observa, diferența se prezintă doar la setul de procente 10-90.

În continuare, se vor ilustra două figuri, pentru a evidenția mai clar, pe grafic, diferențele apărute în urma acestor schimbări minore dintre procente. În figura 2 se va prezenta procesul de învățare, împreună atât cu acuratețea la testare cât și cu pierderile. Procentele acestui set sunt de 90-10.

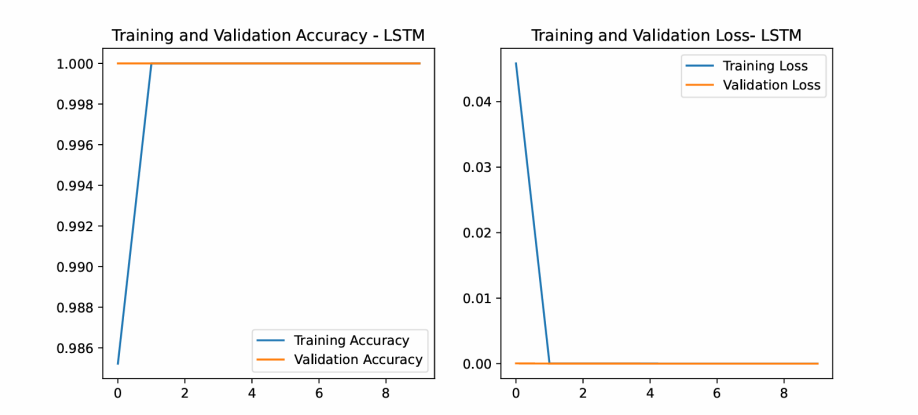

Figure 2: Procesul de învățare și validare, acuratețea, respectiv pierderile, pentru 90-10

În figura 3 se vor prezenta aceleași caracteristici prezentate anterior, însă, procentele de această dată sunt de 10-90.

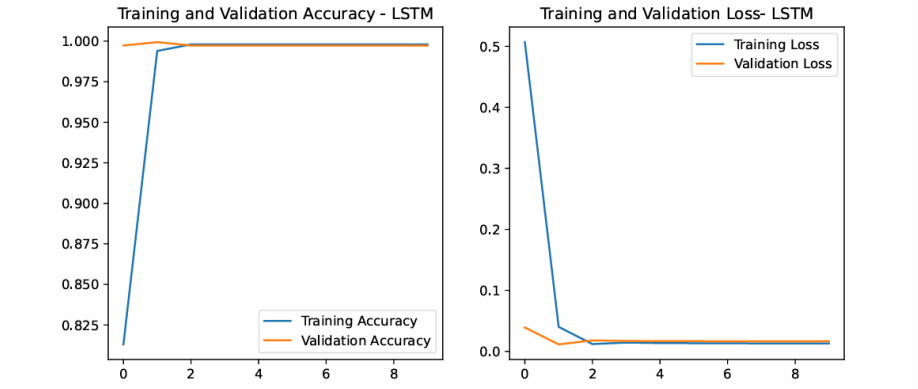

Figure 3: Procesul de învățare și validare, acuratețea, respectiv pierderile, pentru 10-90

De asemenea, se vor prezenta și matricile de confuzie corespunzătoare graficelor. În figura 4 se va afișa matricea de confuzie pentru procentele 90-10.

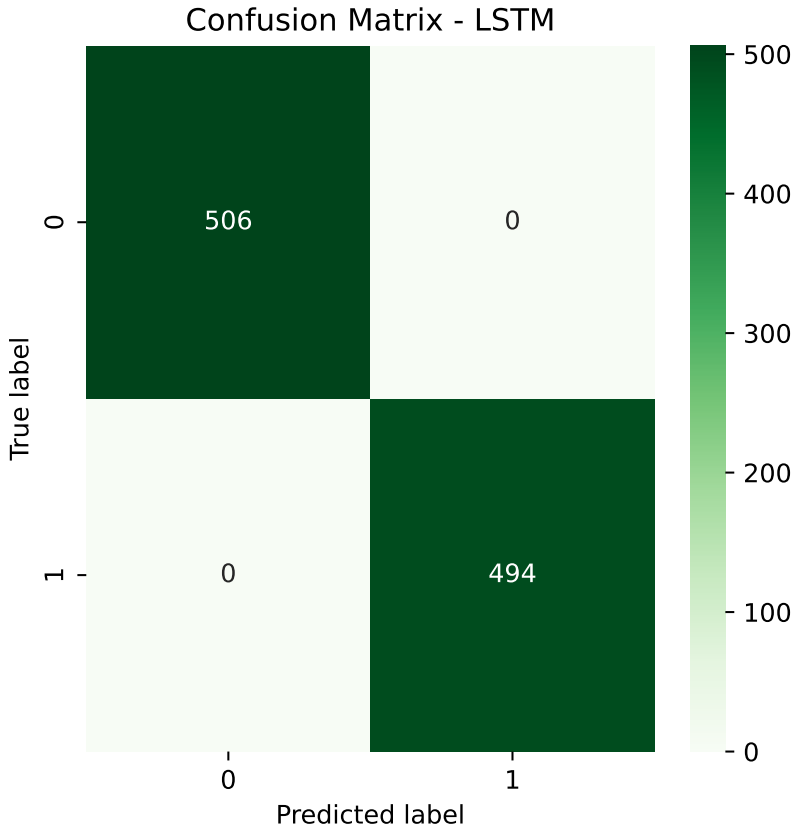

Figure 4: Matricea de confuzie pentru procentele 90-10

Iar în figura de mai jos se va afișa matricea de confuzie pentru procentele 10-90.

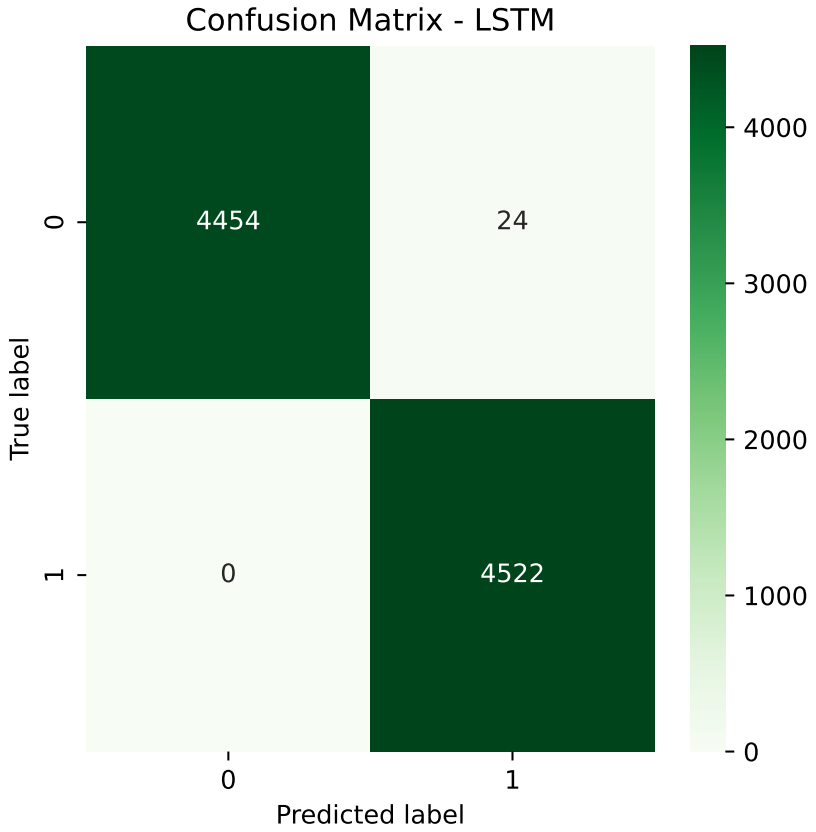

Figure 5: Matricea de confuzie pentru procentele 10-90

Metrici- LSTM

În perspectivă, au fost realizate o serie de analize cu diferite metrici, pentru a permite o comparație între aceste elemente. În figura 6 se vor prezenta metricile folosite pentru a testa modelul LSTM, respectiv: Accuracy, F1-Score și Recall.

Metrics			
Train-Test Size	Accuracy	F1 Score	Recall
90-10	1.0	1.0	1.0
80-20	1.0	1.0	1.0
70-30	1.0	1.0	1.0
60-40	1.0	1.0	1.0
50-50	1.0	1.0	1.0
40-60	1.0	1.0	1.0
30-70	1.0	1.0	1.0
20-80	1.0	1.0	0.9996
10-90	0.997	1.0	0.9996

Figure 6: Testarea modelului LSTM folosind diferite metrici

4.2 Random Forest

În figura 7 se va prezenta acuratețea obținută în urma testării, asemănător cu procesul de la modelul LSTM.

Random Forest	
Train-Test Size	Accuracy
90-10	1.0
80-20	1.0
70-30	1.0
60-40	1.0
50-50	1.0
40-60	1.0
30-70	1.0
20-80	1.0
10-90	1.0

Figure 7: Acuratețea în urma testării cu diferite procente

După cum se observă, rezultatele au rămas neschimbate, acuratețea menținându-se constantă pe parcursul analizei.

În continuare, se va prezenta graficul pentru setul de date având procentele 90-10.

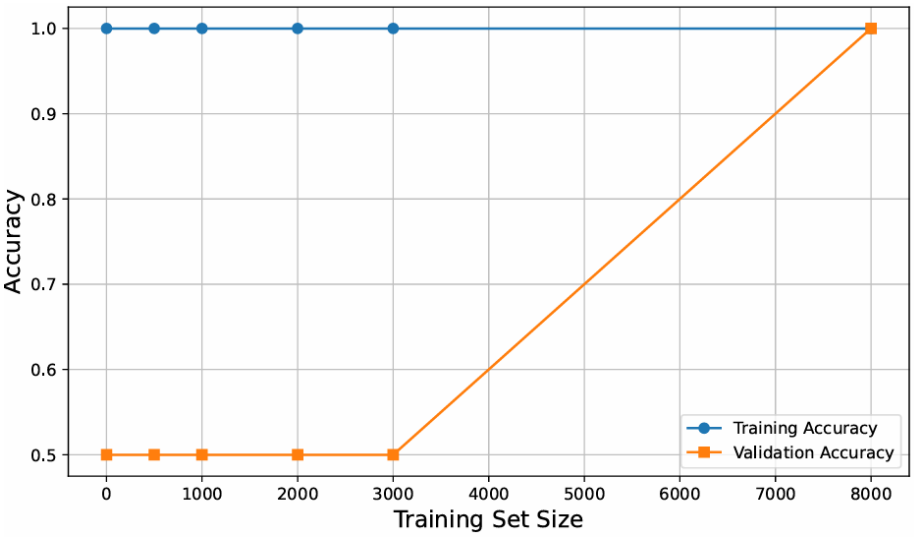

Figure 8: Procesul de învățare și validare, pentru 90-10

Pentru a se putea face o comparație, în figura 11 se va ilustra graficul pentru setul de date dispunând de procentele 10-90.

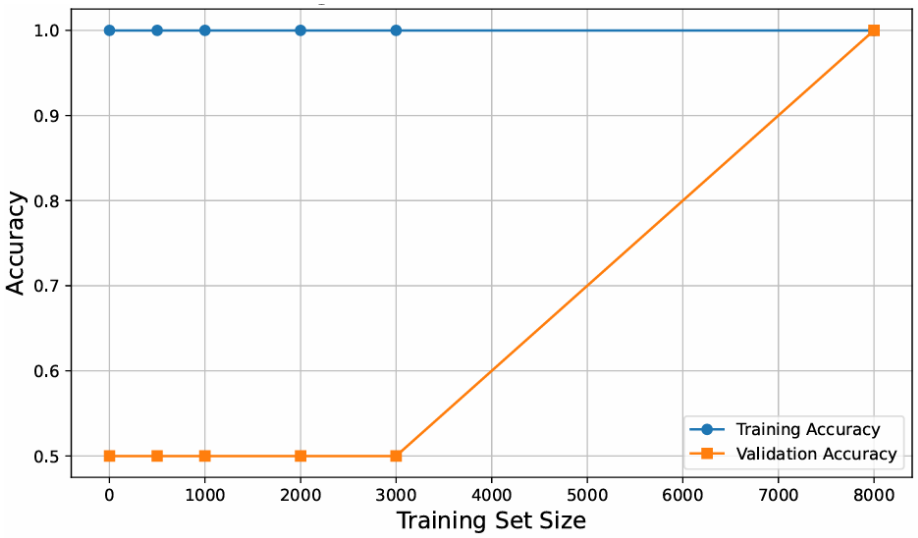

Figure 9: Procesul de învățare și validare, pentru 10-90

De asemenea, mai departe se vor prezenta matricile de confuzie asociate procentelor. În figura de mai jos se va afișa figura conținând procentele 90-10.

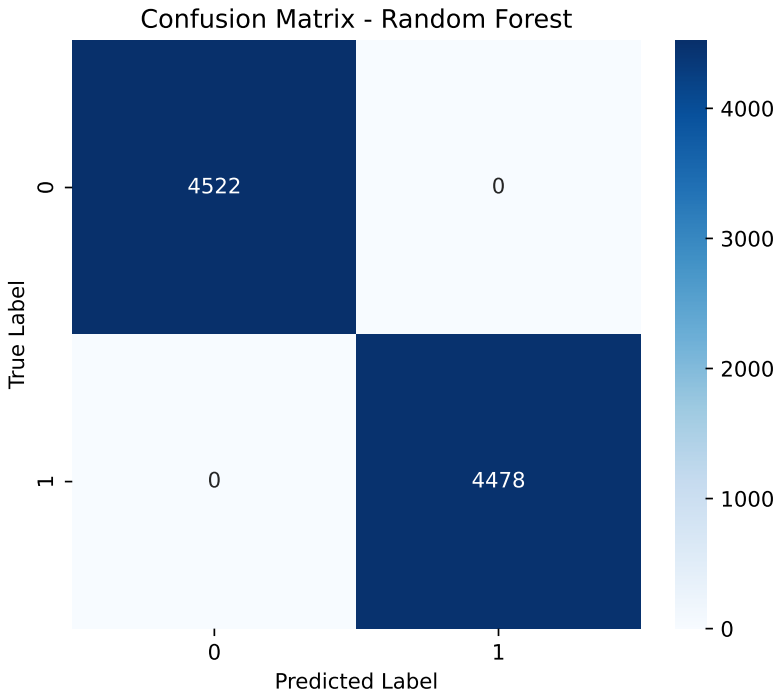

Figure 10: Matricea de confuzie pentru Random Forest, având 90-10

Iar pentru procentele de 10% date de antrenare și 90% date de testare, matricea de confuzie va fi ilustrată în figura 11.

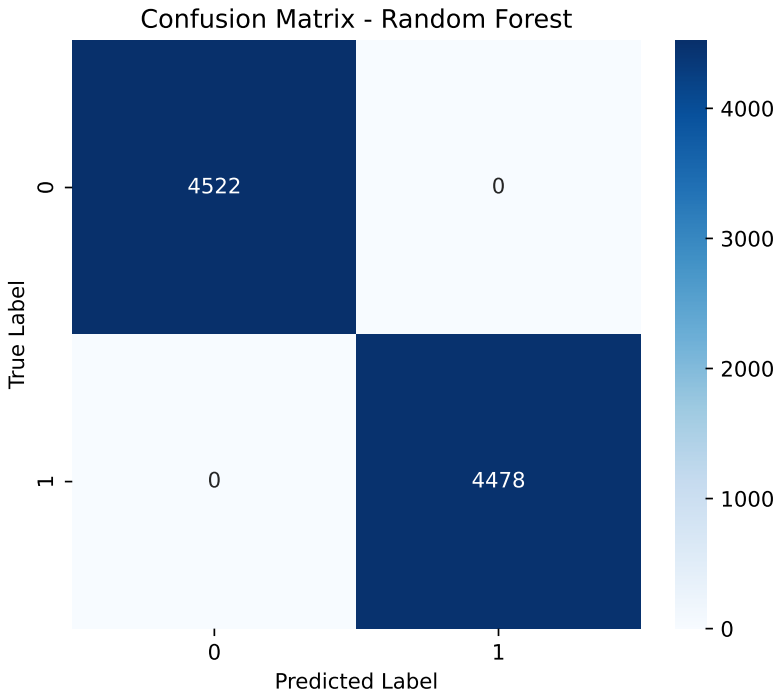

Figure 11: Matricea de confuzie pentru Random Forest pentru 10-90

Metrici - Random Forest

În următoarea parte, se vor prezenta metricile folosite pentru a testa modelul Random Forest, respectiv: Accuracy, F1-Score și Recall. În figura 12 se va ilustra acest lucru.

Metrics			
Train-Test Size	Accuracy	F1 Score	Recall
90-10	1.0	1.0	1.0
80-20	1.0	1.0	1.0
70-30	1.0	1.0	1.0
60-40	1.0	1.0	1.0
50-50	1.0	1.0	1.0
40-60	1.0	1.0	1.0
30-70	1.0	1.0	1.0
20-80	1.0	1.0	1.0
10-90	1.0	1.0	1.0

Figure 12: Metricile modelului Random Forest

4.3 KNN

În figura 13 este prezentată acuratețea obținută la testare pentru toate seturile de date corespunzătoare procentelor menționate, respectiv metricile, de asemenea menționate la LSTM și Random Forest. Datele prezentate în tabel nu sunt constante, însă, acestea variază într-o proporție mai mare față de celalalte două modele: LSTM și Random Forest.

KNN	
Train-Test Size	Accuracy
90-10	0.998
80-20	0.999
70-30	0.999
60-40	0.99925
50-50	0.9992
40-60	0.999333333
30-70	0.99514
20-80	0.99725
10-90	0.9894

Figure 13: Acuratețea în urma testării cu diferite procente

Ulterior, se vor afișa graficele pentru două seturi de date având procentele: primul cuprinzând 90% date de antrenament, 10% date de testare, în figura 14, iar al doilea prezintă 10% date de antrenament și 90% date de testare, ilustrat în figura 15.

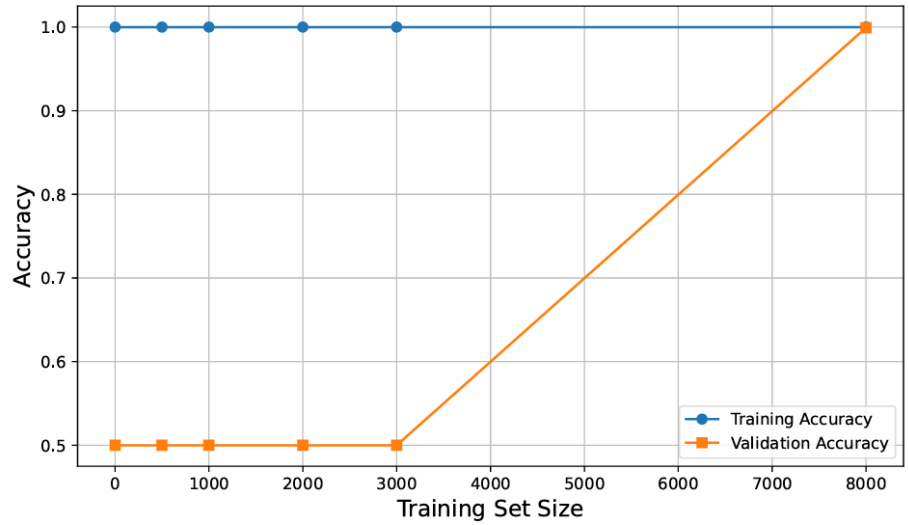

Figure 14: Procesul de învățare și validare, pentru 90-10

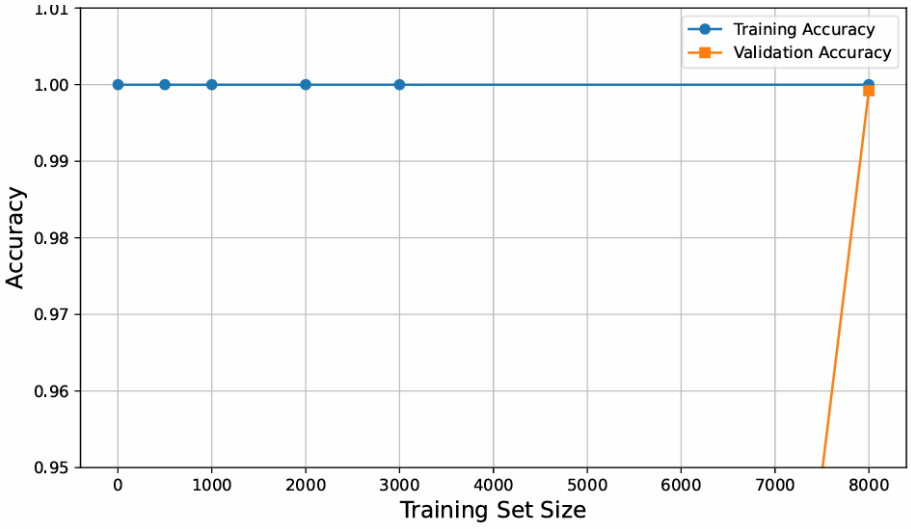

Figure 15: Procesul de învățare și validare, pentru 10-90

Astfel, având această imagine de ansamblu, se pot afișa matricile de confuzie pentru a ilustra mai în detaliu analiza modelului având diferite procente de antrenare și de testare, precum și comparația dintre acestea.

Aceasta este matricea potrivită modelului cu 90% date de antrenament și 10% de testare, iar în figura 17 se va prezenta matricea cu 10% date de antrenament și 90% date de testare.

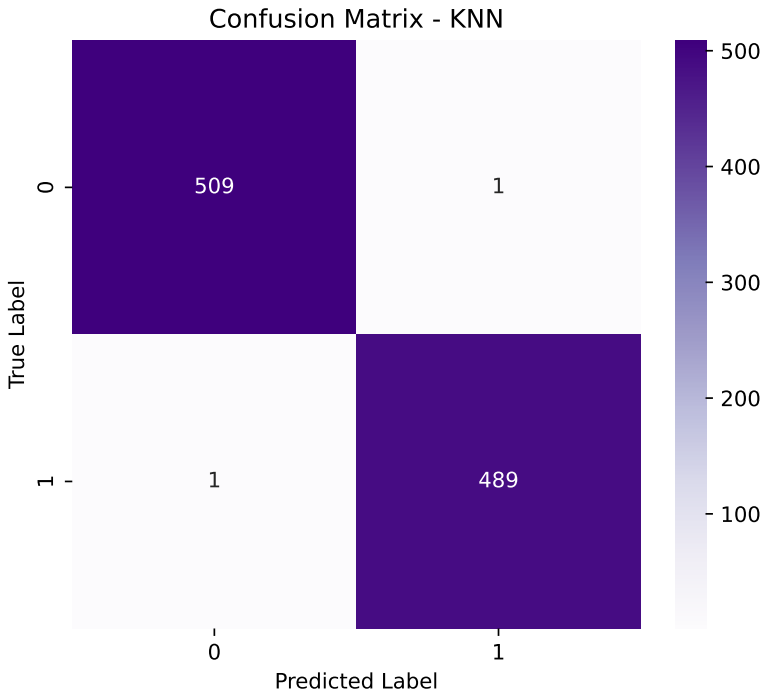

Figure 16: Matricea de confuzie pentru procentele 90-10

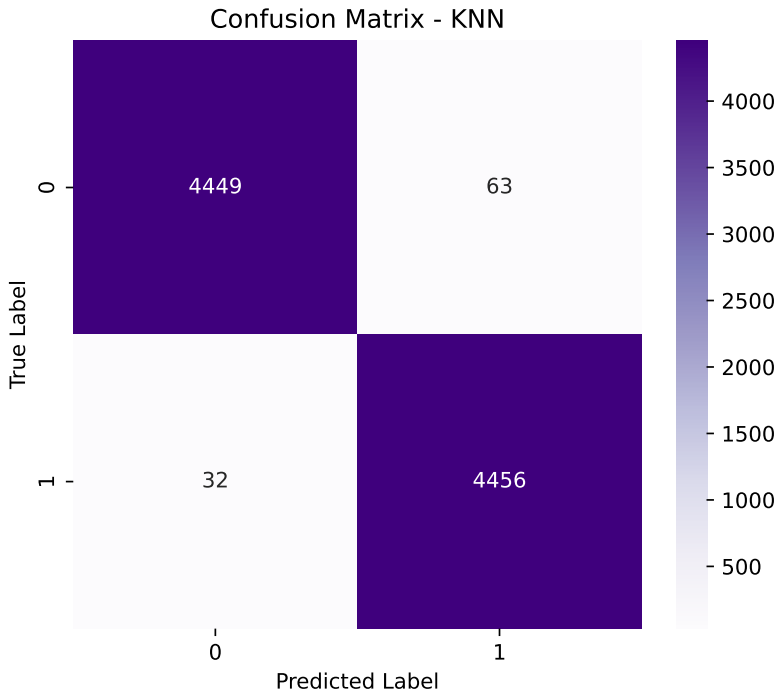

Figure 17: Matricea de confuzie pentru 10-90

Este o diferență semnificativă în rezultatul acestei testări, în comparație cu LSTM și Random Forest.

Metrici- KNN

În această secțiune, se vor prezenta metricile folosite, aceleași metrici precizate la LSTM și Random Forest. În 18 se va ilustra figura reprezentativă pentru aceste metrici.

Metrics			
Train-Test Size	Accuracy	F1 Score	Recall
90-10	1.0	1.0	1.0
80-20	0.99	1.0	1.0
70-30	0.99	1.0	1.0
60-40	0.99	1.0	1.0
50-50	0.99	1.0	1.0
40-60	0.99	1.0	1.0
30-70	0.99	0.99	1.0
20-80	0.99	0.99	1.0
10-90	0.99	0.99	0.99

Figure 18: Metricile modelului KNN

5 Analiza Comparativă a LSTM, Random Forest și KNN

Această secțiune este destinată vizualizării simultane a celor 3 modele, comparându-le performanțele. Graficul realizat ilustrează acuratețea acestora pe parcursul procesului de testare, iar ilustrația 19 ajută la identificarea modelului cel mai performant.

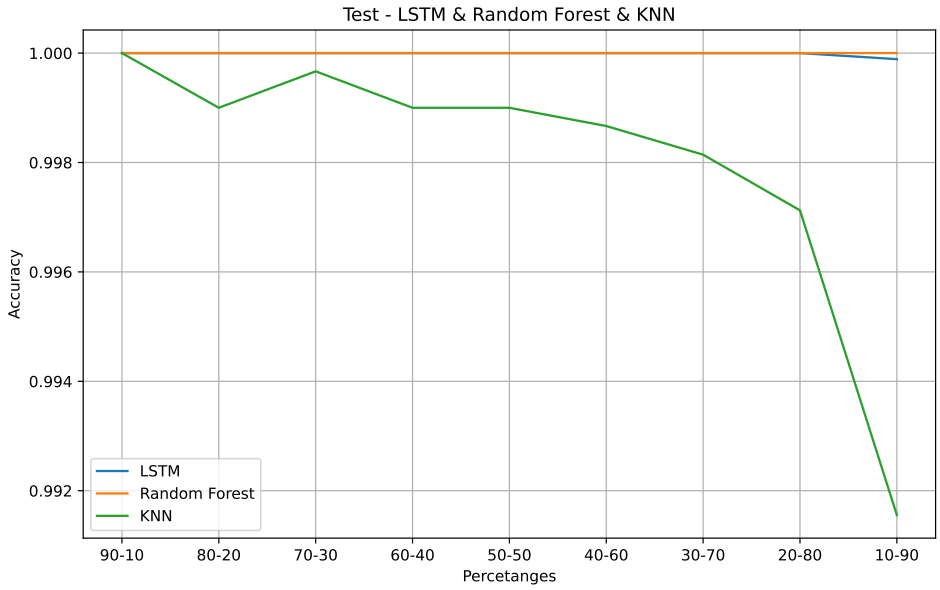

Figure 19: Performanțele modelelor LSTM, Random Forest și KNN

6 Concluzii

În această lucrare s-au analizat tehnicile de învățare automată LSTM, Random Forest și KNN cu scopul de a identifica performanța fiecărui model în detectarea atacurilor de tip Phishing. S-au realizat atât grafice pentru a pune în evidență și a observa mai clar performanțele modelelor, cât și calcule ale metricilor utilizate pentru evaluare, acestea fiind: Accuracy, Recall și F1-score. Rezultatele au fost satisfăcătoare, indicând o performanță bună a modelelor. S-a constatat faptul că modelul KNN a oferit rezultate puțin mai diferite față de celelalte două modele, însă nu nesatisfăcătoare, astfel că, toate 3 modelele au ajuns la o acuratețe de 100%. Pe viitor se vor exploata alte metode de analizare și detectare a atacurilor de tip Phishing.

6.1 Referințe

Referințele folosite pentru acest proiect: [6] [7] [12] [5] [4] [8] [2] [1] [11] [10] [9] [3]

References

[1] 3 methods to save plots as images or pdf files in matplotlib.

[2] how to plot correctly loss curves for training and validation sets?

[3] How to train tensorflow models in python.

[4] How to visualize knn in python.

[5] Impact of dataset size on deep learning model.

[6] SK Hasane Ahammad, E Venkatesh Babu, Sunil D. Kale, Gopal D. Upadhye, Amol V. Dhumane, Sandeep Dwarkanath Pande, and Mr. Dilip Kumar Jang Bahadur. Phishing url detection using machine learning methods. *Advances in Engineering Software*, 173:103288, 2022.

[7] Andronicus A. Akinyelu and Aderemi O. Adewumi. Classification of phishing email using random forest machine learning technique. *Journal of Applied Mathematics*, 2014:1–9, 2014.

[8] Matplotlib Developers. Matplotlib pyplot tutorial.

[9] Stack Exchange. Making sense of a accuracy plot for a 5 fold training using random forest, 2020.

[10] Vahid Shahrivari, Mohammad Mahdi Darabi, and Mohammad Izadi. Phishing detection using machine learning techniques, 2020.

[11] Sruthi. Understanding random forest algorithm with examples.

[12] Shreya Gopal Sundari. Phishing website detection by machine learning techniques.