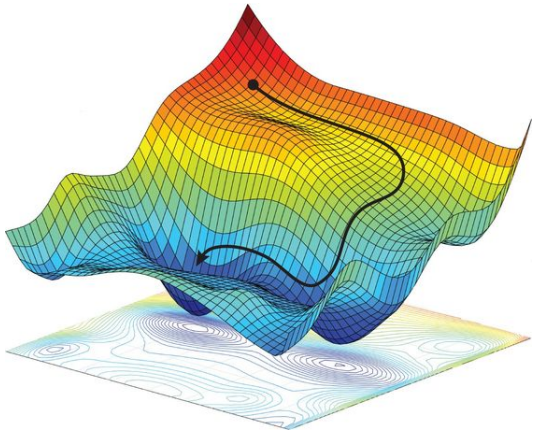
Fakulta elektrotechniky a informatiky



Fakulta elektrotechniky a informatiky

Katedra kybernetiky a umelej inteligencie

Predmet : **Neurónové siete**
kurz 2022 / 2023

Využitie neurónových sietí v detekcii plagiátov

Spracovali:
Orydoroha Bohdan, Roiko Oleksii,
Tsimbota Vladyslav a Zelenska Zlata

KKUI
Katedra kybernetiky a umelej inteligencie

Obsah

1 Úvod

Plagiátorstvo predstavuje použitie cudzej práce alebo myšlienok bez riadneho uznania pôvodného zdroja. Je to považované za nečestný čin, ktorý môže mať vážne dôsledky v akademickom aj profesionálnom prostredí. Plagiátorstvo môže zahŕňať kopírovanie textu, obrázkov, hudby alebo iného vlastníctva bez povolenia a ich prezentovanie ako vlastné dielo.

Plagiátorstvo sa prelína s autorským právom a literárnymi vlastníckymi právami. Autorské právo chráni práva tvorcov a vlastníkov originálnych diel, ako sú knihy, články, hudba a softvér. Keď niekto plagiuje, v podstate porušuje tieto práva tým, že používa dielo bez súhlasu vlastníka a bez riadneho uvedenia zdroja.

Napríklad, študent môže plagiovať skopírovaním a vložením odseku z publikovaného článku do svojej eseje bez uvedenia zdroja. Rovnako by mohol spisovateľ plagiovať začlenením častí románu niekoho iného do svojho vlastného diela bez povolenia alebo uvedenia zdroja. V oboch prípadoch plagiátorstvo porušuje práva vlastníka autorských práv a môže viesť k právnym následkom.

V histórii sa vyskytlo niekoľko slávnych príkladov plagiátorstva, ako napríklad v prípade vynálezcu a vedca Nikolu Teslu. Tesla je známy svojimi významnými príspevkami k vývoju elektrických systémov striedavého prúdu, ktoré tvoria základ modernej distribúcie elektriny. Čelil však tvrdej konkurencii iných vynálezcov, vrátane Thomasa Edisona, ktorý bol zástancom jednosmerných elektrických systémov[1].

Teslova práca bola často zatienená prácou iných vynálezcov a nie vždy bol za svoje príspevky ocenený. V roku 1888 bolo Teslovi vydaných niekoľko patentov na jeho indukčný motor na striedavý prúd, ktorý bol v tom čase prelomom v elektrotechnike. Nebol to však Tesla, kto sa preslávil týmto vynálezom, ale skôr George Westinghouse, americký podnikateľ a inžinier, ktorý kúpil Teslove patenty a komercializoval technológiu[1]. Hoci Tesla nakoniec získal uznanie za svoju prácu, zatienenie jeho vynálezov inými vynálezcami a obchodníkmi je príkladom toho, ako môže plagiátorstvo a nedostatok správneho prisúdenia ovplyvniť aj tie najbrilantnejšie mysle.

V posledných rokoch technológie zohrávajú významnú úlohu pri odhaľovaní a prevencii plagiátorstva. Jedným z najsľubnejších prístupov v tejto oblasti je aplikácia neurónových sietí na identifikáciu a analýzu potenciálne plagiovaného obsahu. V nasledujúcich častiach preskúmame koncept neurónových sietí, ich aplikáciu pri odhaľovaní plagiátov a ako môžu pomôcť zlepšiť presnosť a efektívnosť identifikácie plagiátu.

2 Špecifikácia oblasti aplikácie a popis problému

Použitie neurónových sietí pri detekcii plagiátov prináša niekoľko výhod a činí ich atraktívnou voľbou pre túto úlohu. Nasledujú niektoré kľúčové argumenty podporované výskumnými citáciami:

- Vysoká presnosť – neurónové siete, najmä modely hlbokého učenia, sa osvedčili ako veľmi presné pri identifikácii vzorov a korelácií v rámci veľkých súborov údajov. Štúdie preukázali účinnosť modelov neurónových sietí pri identifikácii podobností a rozdielov v texte, čím poskytli robustný a spoľahlivý spôsob detekcie plagiátorstva[2].

- Sémantické porozumenie – neurónové siete, najmä tie založené na transformátorových architektúrach ako BERT, preukázali veľký úspech v pochopení sémantického významu text[3]. Táto schopnosť im umožňuje detegovať nielen presné zhody, ale aj parafrazovaný alebo prepísaný obsah, čo je kľúčové pre efektívnu detekciu plagiátov.

- Adaptabilita – neurónové siete sa dokážu učiť a prispôsobovať novým údajom a meniacim sa trendom v jazyku[4]. Táto prispôsobivosť ich robí vhodnými pre úlohu detekcie plagiátov, keďže môžu byť trénované na rozpoznávanie rôznych foriem a štýlov písania a prispôsobovať sa zmenám jazykových noriem.

- Škálovateľnosť – detekcia plagiátov často vyžaduje spracovanie veľkého množstva textu. Neurónové siete je možné paralelizovať a implementovať na grafických procesoroch (GPU), čo umožňuje efektívne spracovanie rozsiahlych súborov údajov[5].

Zhrnutím, použitie neurónových sietí pri detekcii plagiátov je podporované rôznymi štúdiami, ktoré dokazujú ich vysokú presnosť, sémantické porozumenie, adaptabilitu a škálovateľnosť, čo z nich robí dobrú voľbu na riešenie tejto výzvy. Okrem ich schopnosti adaptovať sa na nové vzory a štýly textu môžu neurónové siete rýchlo a automaticky skenovať veľké množstvo textov, čo umožňuje identifikáciu prípadov plagiátorstva v reálnom čase. S pokrokom v oblasti hlbokého učenia a rozvojom nových techník v analýze textu sa očakáva, že presnosť a účinnosť detekcie plagiátorstva pomocou neurónových sietí sa bude ďalej zlepšovať.

3 Prehľad aplikácií neurónových sietí

Pri detekcii plagiátorstva sa neurónové siete trénujú na veľkých množstvách textových dát, aby sa naučili rozpoznávať podobnosti medzi rôznymi textami. Tieto siete môžu identifikovať plagiát v dvoch hlavných formách: parafrázovanie a patchwork. Parafrázovanie zahŕňa prepísanie textu s použitím iných slov, zatiaľ čo patchwork zahŕňa kombináciu viacerých zdrojov do jedného textu[5].

Pri analýze textu neurónové siete rozdeľujú text na menšie časti, ako sú vety alebo slová, a porovnávajú ich s veľkým množstvom existujúcich textov v databáze. V prípade, že neurónová sieť identifikuje vysokú mieru podobnosti medzi analyzovaným textom a niektorým zo zdrojových textov, môže byť príslušná časť textu označená ako potenciálne plagiovaná. Tento proces sa môže vykonávať rýchlo a automaticky, čo umožňuje efektívne skenovanie veľkého množstva textov a identifikáciu prípadov plagiátorstva[6].

Neurónové siete, konkrétne hlboké učenie a rekurentné neurónové siete (RNN), sa ukázali ako efektívne v detekcii plagiátorstva vďaka svojej schopnosti extrahovať a porovnávať črty z textových dát. Pre lepšie pochopenie toho, ako neurónové siete fungujú pri detekcii plagiátorstva, si rozoberme niektoré z kľúčových konceptov a techník:

- Vektorová reprezentácia textu – pre spracovanie textu neurónovými sieťami je potrebné previesť slová alebo vety do numerických vektorov. Táto prevodná technika sa nazýva embedding a vytvára vektorový priestor, kde sú podobné slová alebo frázy zoskupené bližšie k sebe. Medzi bežné metódy embeddingu patrí Word2Vec, GloVe alebo BERT[7].
- Rekurentné neurónové siete (RNN) – RNN sú špeciálne navrhnuté pre prácu s časovými radmi alebo sekvenciami, ako sú textové dáta[8]. RNN sú schopné zachytiť kontext a závislosti medzi slovami alebo vetami v texte, čo je kľúčové pre analýzu podobnosti medzi dvoma textami. RNN môžu byť rozšírené o LSTM (Long Short-Term Memory) alebo GRU (Gated Recurrent Unit) vrstvy, ktoré zlepšujú schopnosť siete zachovať si dlhodobé závislosti v sekvenciách.
- Siamské neurónové siete – tento typ neurónových sietí sa používa na porovnávanie dvoch alebo viacerých vstupov, aby sa zistilo, či sú podobné alebo nie. Siamské siete sú trénované na pároch podobných a odlišných vstupov, aby sa naučili rozpoznávať podobnosti a rozdiely medzi textami. Tieto siete sa často používajú na identifikáciu parafrázovania alebo patchworku v potenciálne plagiovaných textoch[9].
- Konvolučné neurónové siete (CNN) – CNN sú pôvodne navrhnuté pre analýzu obrazu, ale môžu sa tiež účinne použiť na analýzu textu. Konvolučné vrstvy v sieti extrahujú miestne črty zo vstupného textu, napríklad n-gramy alebo časté slovné spojenia. Tieto črty sa potom použijú na identifikáciu podobnosti medzi analyzovaným textom a textami v databáze[10]. Konvolučné neurónové siete môžu byť kombinované s rekurentnými neurónovými sieťami (RNN) alebo siamskými neurónovými sieťami pre ešte presnejšiu analýzu textových dát.

Po úspešnom trénovaní a validácii sa neurónová sieť môže použiť na analýzu nových, neoznačených textov. V prípade, že neurónová sieť identifikuje vysokú mieru podobnosti medzi analyzovaným textom a niektorým zo zdrojových textov, môže byť príslušná časť textu označená ako potenciálne plagiovaná. Tieto identifikované časti textu sa potom môžu preveriť manuálne, aby sa zabezpečilo, že ide skutočne o plagiátorstvo.

4 Prípadová štúdia

Aby sme sa mohli ponoriť ešte hlbšie do práce NN vo svete plagiátorstva, skonštruovali sme platný príklad neurónovej siete. Naša neurónová sieť sa pozrie na dve otázky a pokúsi sa určiť, či majú tieto dve otázky rovnaký význam (Ide o rovnaký koncept, aký používa väčšina moderných detektorov plagiátov. Porovnávajú sa celé vety alebo niekedy odseky textu, aby sa zistilo, či sú sémanticky totožné). Na tento účel sme si vybrali model RNN, ktorý je na túto úlohu mimoriadne vhodný vďaka svojej schopnosti spracovávať sekvenčné údaje, ako je text. Použili sme variant LSTM, pretože je lepšie vybavený na spracovanie dlhodobých závislostí v texte. V našom prípade (porovnávanie dvoch viet) sa dôležité informácie môžu nachádzať ďaleko od seba v kontexte jednej vety, čo by bolo ťažké pre obyčajný RNN výzvou.

Všetko sa začína spracovaním údajov. Na vstupe máme súbor údajov obsahujúci 10000 dvojíc otázok a skóre zmysluplnosti (či sú dvojice zmysluplne rovnaké alebo nie). Vety sú predspracované (text vo vetách je písaný všetkými malými písmenami a potom sú vety rozdelené na zoznam slov (tokenov)). Kombinácie dvoch viet (vo forme spracovaných zoznamov) sa vložia do modelu Word2Vec (Model Word2Vec zahŕňa učenie vložených slov, ktoré dokážu zachytiť sémantické vzťahy medzi slovami na základe ich bežného použitia v textovom korpuse). Word2Vec nám umožňuje zistiť, aké silné sú väzby medzi slovami (korelačné koeficienty). Na zisťovanie vložených slov v kontexte Word2Vec sa používa Skip-gram: na základe štúdie [13] má najlepšie parametre. Okrem toho je potrebné vyrovnať počet slov vo vetách, t. j. pridať "prázdne miesta"(nulové hodnoty) na koniec menšej vety. Robí sa to preto, aby sa zabezpečilo, že matice, ktoré sa násobia v NS, majú rovnakú dĺžku.

Ako trenovacie data sme pouzili 80% nasho datasetu a pre testovanie a vyhodnotenie 20%. Po spracovaní všetkých údajov naučil sa náš model LSTM. LSTMModel je jednovrstvová sieť LSTM, po ktorej nasleduje lineárna vrstva. Vrstva LSTM má vstup s veľkosťou 128 (veľkosť vloženia Word2Vec) a má veľkosť skrytej vrstvy 128. Learning rate je nastaveny na 0,001, s počtom epoch 15 a batch size 32 jednotiek. Ako optimalizátor sme zvolili Adam(naprieč ostatným je konzistentný). Aktivačná funkcia použitá v tomto prípade je sigmoid, aby na konci sme malí pravdepodobnosti. Ako loss function sme zvolili Binary Cross Entropy, pretože máme iba dva možné výstupy (rovnaké vety alebo rôzne). Rozdiel medzi výstupmi pre dve otázky nám indikuje pravdepodobnosť duplicity otázok. Vďaka backpropagation sme schopní efektívne upravovať váhy v našom modeli.

Model LSTM

```
class LSTMModel(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers):
        super(LSTMModel, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, 1)

    def forward(self, x):
        _, (hidden, _) = self.lstm(x)
        output = self.fc(hidden[-1])
        return torch.sigmoid(output)
```

Spracovanie dát

```
url = "https://raw.githubusercontent.com/S4IKEz/stuff/main/questions1.csv"
data = pd.read_csv(url, usecols=['question1', 'question2', 'is_duplicate'])

data['question1'] = data['question1'].str.lower().str.split()
data['question2'] = data['question2'].str.lower().str.split()

# Trénujte model Word2Vec
sentences = data['question1'].tolist() + data['question2'].tolist()
model_w2v = Word2Vec(sentences, vector_size=128, window=5, min_count=1, workers=4)

# Encode Word2Vec
def encode_questions(question):
    return np.array([model_w2v.wv[word] for word in question])

data['q1_encoded'] = data['question1'].apply(encode_questions)
data['q2_encoded'] = data['question2'].apply(encode_questions)

X1 = pad_sequences(data['q1_encoded'].tolist())
X2 = pad_sequences(data['q2_encoded'].tolist())
y = data['is_duplicate'].values

X1_train, X1_test, X2_train, X2_test, y_train, y_test =
train_test_split(X1, X2, y, test_size=0.2, random_state=42)
```

Parametre

```
input_size = 128
hidden_size = 128
num_layers = 1
num_epochs = 15
learning_rate = 0.001
batch_size = 32
```

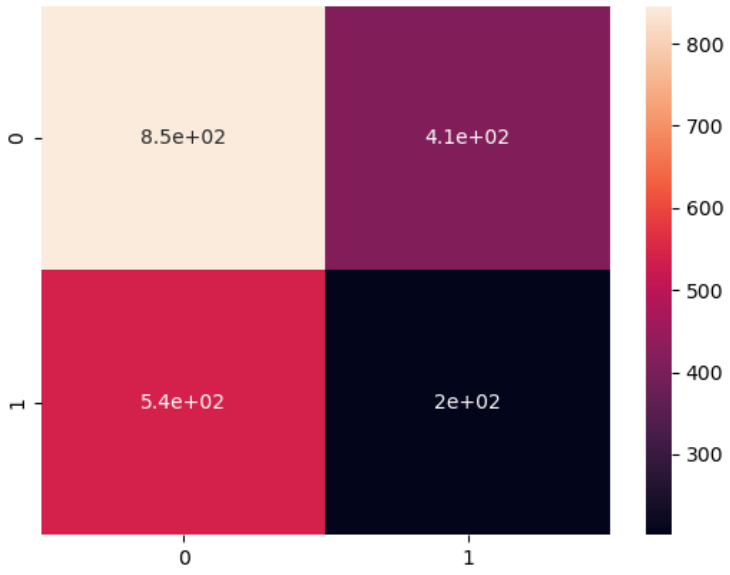
Trenovanie

```
model = LSTMModel(input_size, hidden_size, num_layers)
criterion = nn.BCELoss()
optimizer = optim.Adam(model.parameters(), lr=learning_rate)

model.train()
for epoch in range(num_epochs):
    for i, (q1, q2, labels) in enumerate(train_loader):
        q1_out = model(q1)
        q2_out = model(q2)
        # print(q1_out, q2_out);
        out = torch.abs(q1_out - q2_out)

        loss = criterion(out, labels)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

Vysledky

```
model.eval()
with torch.no_grad():
    X1_test_tensor = torch.tensor(X1_test, dtype=torch.float32)
    X2_test_tensor = torch.tensor(X2_test, dtype=torch.float32)
    q1_test_out = model(X1_test_tensor)
    q2_test_out = model(X2_test_tensor)
    test_out = torch.abs(q1_test_out - q2_test_out)
    test_preds = (test_out > 0.5).type(torch.float32).view(-1)
    y_true = torch.tensor(y_test, dtype=torch.float32).view(-1)
    y_pred = test_preds
    cm = confusion_matrix(y_true, y_pred)
    print("Confuzna matica:")
    print(cm)
    test_acc = torch.mean((test_preds == torch.tensor(y_test, dtype=torch.float32))
    .type(torch.float32)).item()
    print(f"Presnost: {test_acc}")
```

Po natrénovaní modelu ho otestujeme a overíme výkon našej NN pomocou konfuznej matice. Celkovým výsledkom bude percentuálny podiel správne klasifikovaných vzoriek.



Obr. 1: Konfúzna matica

5 Diskusia

V tejto časti preskúmame zistenia uvedenej štúdie a podrobnejšie sa pozrieme na kľúčové kritériá porovnávania. Po prvé, aplikácia vektorovej reprezentácie na text transformuje text na zoznam vektorov, ktoré udržiavajú sémantické a syntaktické vlastnosti poskytnuté algoritmami hlbokého učenia. Po druhé, kritérium úrovne spracovania určuje, či je text spracovaný na úrovni slov alebo viet. A napokon, metóda podobnosti sa týka prístupov použitých na výpočet podobnosti medzi vektormi, ktoré reprezentujú texty, a poskytuje prehľad o silných a slabých stránkach jednotlivých metód.

Väčšina prístupov používa na transformáciu vektorov metódu Word2Vec alebo Doc2Vec, pričom na zachovanie sémantického aspektu daného textu je najúčinnejšou metódou mikolovská reprezentácia. Transformácia textu na zoznam viet je najvhodnejšou reprezentáciou, pretože zohľadňuje význam textu.

Čo sa týka metód použitých na výpočet podobnosti: na určenie, či medzi analyzovanými textami existuje podobnosť, sa využívajú rôzne prístupy. Mnohé z týchto prístupov využívajú v svojej architektúre CNN a RNN, avšak väčšina z nich sa spolieha na vektorovú reprezentáciu na úrovni slov. Toto obmedzenie znižuje ich schopnosť zisťovať podobnosti medzi celými vetami alebo textami, namiesto toho sa sústredia iba na slová.[11]

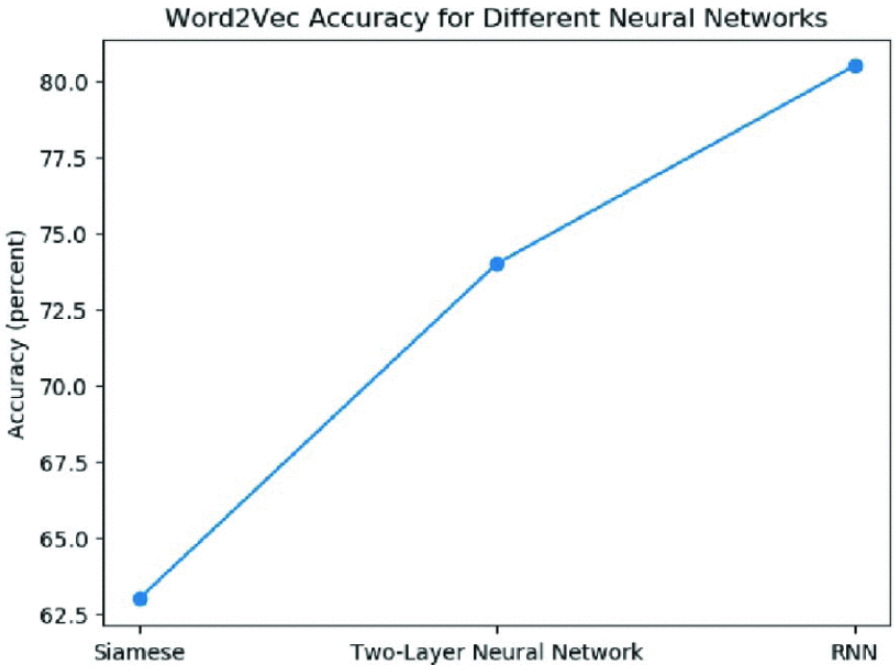
Takmer všetky prístupy používajú na výpočet podobnosti medzi dokumentmi kosínus, ktorý sa vykonáva po slovách alebo po vetách. Táto metóda však môže viesť k nespoľahlivým výsledkom, pretože dva dokumenty môžu mať rovnaké slovo alebo vetu bez toho, aby boli sémanticky podobné. Na riešenie tohto problému je potrebný prístup, ktorý reprezentuje text ako zoznam viet, ktorý sa transformuje na zoznam vektorov. Spracovanie aplikované na zoznam viet by malo zachovať sémantický aspekt textu a na zisťovanie podobnosti sa môže použiť algoritmus ako RNN.[12]

Ďalšou výhodou je, že neurónové siete sa dajú použiť v rôznych odvetviach a kontextoch. Môžu sa používať v akademickom prostredí na kontrolu študentských prác, či nie sú plagiátorské, vo vydavateľskom priemysle na kontrolu duplicitného obsahu a dokonca aj v právnom odvetví na identifikáciu prípadov porušenia autorských práv.

Na používanie neurónových sietí pri odhaľovaní plagiátov je k dispozícii niekoľko nástrojov a platforiem. Medzi najobľúbenejšie patria Turnitin, iThenticate a PlagScan. Tieto platformy umožňujú používateľom nahrať dokumenty a analyzovať ich na potenciálne prípady plagiátorstva.

6 Zhodnotenie a záver

Plagiátorstvo sa v súčasnosti stáva čoraz častejším problémom, najmä s príchodom technológií, ako je ChatGPT a ďalšie. Je to nepríjemný pocit, keď sa niekto pokúša ukradnúť váš nápad alebo ho nelegálne skopírovať. Preto sme sa rozhodli napísať tento článok, aby sme zdôraznili význam používania neurónových sietí v boji proti plagiátorstvu. Pomocou RNN a metódy Word2Vec sme dosiahli presnosť detekcie plagiátov 61%, čo je podľa nás slušný výsledok.
Nižšie je uvedený graf "Presnosť metódy Word2Vec pre rôzne NN":



Obr. 2: Presnosť Word2Vec pre rôzne neurónové sietí[13]

Na záver možno povedať, že využitie neurónových sietí pri odhaľovaní plagiátov predstavuje účinný a efektívny spôsob boja proti neoprávnenému kopírovaniu v rôznych doménach. S narastajúcou dostupnosťou digitálnych údajov sa stáva použitie neurónových sietí čoraz populárnejším v mnohých odvetviach. Preto je kľúčové, aby jednotlivci a organizácie sa oboznámili s touto technológiou a začlenili ju do svojich činností s cieľom zaručiť integritu a autenticitu svojho obsahu.

Zoznam použitej literatúry

[1] Jonnes, J. (2004). Empires of Light: Edison, Tesla, Westinghouse, and the Race to Electrify the World. Random House Trade Paperbacks.

[2] Ali, A., & Taqa, A. Y. (2022). Analytical Study of Traditional and Intelligent Textual Plagiarism Detection Approaches. Journal of Education and Science, 31(1), 8-25.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[4] El-Rashidy, M. A., Mohamed, R. G., El-Fishawy, N. A., & Shouman, M. A. (2022). Reliable plagiarism detection system based on deep learning approaches. Neural Computing and Applications, 34(21), 18837-18858.

[5] Gharavi, E., Veisi, H., & Rosso, P. (2020). Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: no training phase. Neural Computing and Applications, 32, 10593-10607.

[6] Alzahrani, S.M., Salim, N., & Abraham, A. (2012). Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42, 133-149.

[7] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: a comprehensive review. ACM computing surveys (CSUR), 54(3), 1-40.

[8] Kulkarni, S., Govilkar, S., & Amin, D. (2021, May). Analysis of Plagiarism Detection Tools and Methods. In Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021).

[9] Tian, Z., Wang, Q., Gao, C., Chen, L., & Wu, D. (2020). Plagiarism detection of multi-threaded programs via siamese neural networks. IEEE Access, 8, 160802-160814.

[10] Benabbou, F. (2020). A New Online Plagiarism Detection System based on Deep Learning. International Journal of Advanced Computer Science and Applications, 11(9).

[11] Shaojie Bai, J. Zico Kolter, Vladlen Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271v2 [cs.LG] 19 Apr 2018

[12] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043.

[13] Hunt, E., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., ... & Oh, P. (2019, November). Machine learning models for paraphrase identification and its applications on plagiarism detection. In 2019 IEEE International Conference on Big Knowledge (ICBK) (pp. 97-104). IEEE.