IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.1120000

Fine-Tuned Understanding: Enhancing Social Bot Detection with Transformer-based Classification

AMINE SALLAH[1], EL ARBI ABDELLAOUI ALAOUI[2], SAID AGOUJIL[3], MUDASIR AHMAD WANI [4], MOHAMED HAMMAD[4,5], AHMED A. ABD EL-LATIF[4,6,7], (Senior Member, IEEE), and YASSINE MALEH[8], (Senior Member, IEEE)

[1]Department of Computer Science, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Errachidia, Morocco (e-mail: a.sallah@edu.umi.ac.ma)
[2]Department of Sciences, Ecole Normale Supérieure, Moulay Ismail University of Meknes, Morocco (e-mail: abdellaoui.e@gmail.com)
[3]École Nationale de Commerce et de Gestion, Moulay Ismail University of Meknes, El Hajeb, Morocco (e-mail: s.agoujil@umi.ac.ma)
[4]EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia (e-mail: mwani@psu.edu.sa; mhammad@psu.edu.sa; aabdellatif@psu.edu.sa)
[5]Department of Information Technology, Faculty of Computers and Information, Menoufia University, Shibin El Kom 32511, Egypt (e-mail: mhammad@psu.edu.sa)
[6]Center of Excellence in Quantum and Intelligent Computing, Prince Sultan University, Riyadh 11586, Saudi Arabia (e-mail: aabdellatif@psu.edu.sa)
[7]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt (e-mail: aabdellatif@psu.edu.sa)
[8]Laboratory laSTI, ENSAK, USMS University, Beni Mellal, Morocco (e-mail: yassine.maleh@ieee.org)

Corresponding author: Yassine Maleh (e-mail: yassine.maleh@ieee.org).

ABSTRACT In recent years, the proliferation of online communication platforms and social media has given rise to a new wave of challenges, including the rapid spread of malicious bots. These bots, often programmed to impersonate human users, can infiltrate online communities, disseminate misinformation, and engage in various activities detrimental to the integrity of digital discourse. It is becoming more and more difficult to discern a text produced by deep neural networks from that created by humans. Transformer-based Pre-trained Language Models (PLMs) have recently shown excellent results in challenges involving natural language understanding (NLU). The suggested method is to employ an approach to detect bots at the tweet level by utilizing content and fine-tuning PLMs, to reduce the current threat. Building on the recent developments of the BERT (Bidirectional Encoder Representations from Transformers) and GPT-3, the suggested model employs a text embedding approach. This method offers a high-quality representation that can enhance the efficacy of detection. In addition, a Feedforward Neural Network (FNN) was used on top of the PLMs for final classification. The model was experimentally evaluated using the Twitter bot dataset. The strategy was tested using test data that came from the same distribution as their training set. The methodology in this paper involves preprocessing Twitter data, generating contextual embeddings using PLMs, and designing a classification model that learns to differentiate between human users and bots. Experiments were carried out adopting advanced Language Models to construct an encoding of the tweet to create a potential input vector on top of BERT and their variants. By employing Transformer-based models, we achieve significant improvements in bot detection F1-score (93%) compared to traditional methods such as Word2Vec and Global Vectors for Word Representation (Glove). Accuracy improvements ranging from 3% to 24% compared to baselines were achieved. The capability of GPT-4, an advanced Large Language Model (LLM), in interpreting bot-generated content is examined in this research. Additionally, explainable artificial intelligence (XAI) was utilized alongside transformer-based models for detecting bots on social media, enhancing the transparency and reliability of these models.

INDEX TERMS BERT, Online Social Networks, NLP, Transfer Learning, Bot Classification, Transformers, Pre-trained Language Models, Explainability, LLM-based prompting.

I. INTRODUCTION

Since bots pose a greater threat to free will and opinion, the problem of bot identification on Twitter has grown in interest as the social network becomes more widely used. Due to the rising complexity of bots, which goes beyond the traditional study of individual features, researchers have concluded that

it is necessary to classify them at a behavioral level on the platform. Numerous individuals worldwide now rely heavily on social media platforms as their primary source of information. Twitter is widely recognized as the most renowned microblogging platform. The usage of bots (@HundredZeros or @TayTweets) is one example of social media manipulation. User accounts that are managed by software algorithms instead of human users are commonly referred to as bot accounts. Bots are programmed to perform specific tasks or actions on digital platforms, often to automate processes or provide certain functionalities. These accounts can interact with other users, generate content, or perform actions based on predefined instructions. The sophistication of current social media bots ranges widely; some are relatively basic, primarily engaging in retweeting content they find interesting, whilst others are more complex and may communicate with actual users. Twitter is one of these platforms that facilitates the rapid dissemination of information throughout its user community. In the field of text generation, the latest neural language models have reached a state where their output is remarkably grammatically accurate, fluent, and coherent. As a result, it has become challenging to differentiate between text generated by these models and text written by humans [1]. Consequently, there is a growing need to explore the effectiveness of existing detection methods proposed in the research literature to distinguish human-generated text from text generated by neural language models. This is especially crucial due to emerging evidence suggesting that humans find this task exceedingly difficult [2].

Twitter postings are character-restricted writings that are limited to 280 characters. Because short messages authored by bots are harder to discern from human-generated texts than lengthier texts, this format is perfect for text-generating algorithms [3]. There has been a significant increase in academic curiosity and study given to identifying and detecting social media bots in recent years. The growing engagement and resultant effect of these automated accounts on numerous social media platforms are driving this increased emphasis [4]. According to statistics published in March 2023 [5], the most recent statistics from an internal study of Twitter bot percentages revealed that fewer than 5% of its users are fraudulent or spam bots. The objective is to create a robust model capable of producing cutting-edge bot detection findings. We investigated various standard word embedding approaches in this context, including Word2Vec [6] and Global Vectors for Word Representation (Glove) [7]. In this study, we explore and empirically assess the performance of pre-trained word embeddings and language models tailored for text analysis from social media. Upon evaluating the pre-trained language models, we proceeded with additional experiments involving pre-trained embeddings sourced from social media and Google news. For this classification task, we selected one prevalent neural architecture in NLP: Long Short Term Memory networks (LSTM) [8].

Advancements in natural language generation (NLG) have led to the development of deep learning models like Generative Pre-trained Transformer 3 (GPT-3) [9], which are capable of generating synthetic text or deep fake text with a high level of linguistic quality. GPT-3 belongs to the Transformer family of language models (LMs), which are known for their ability to capture complex patterns in text data. Recently, transformer-based models have achieved remarkable success in establishing new benchmarks for various natural language understanding (NLU) tasks, including document classification [10], [11], text summarization [12], [13], machine translation [14], [15], and question answering [16]. The achievements in utilizing pre-trained language models (PLMs) have also been replicated in the clinical and biomedical domains. This has been accomplished by training PLMs on extensive clinical or biomedical datasets and subsequently fine-tuning them for various downstream tasks specific to the clinical or biomedical field [17]. The deep learning methodology obviates the necessity for laborious feature engineering, as the model possesses the capacity to autonomously discern and establish the connection between subjects and texts after being trained on an extensive corpus of data. This characteristic aligns with the cognitive reasoning capabilities exhibited by humans [18].

Deep contextualized language models trained via masked language modeling, such as Bidirectional Encoder Representations from Transformers (BERT) [19], have shown advanced performance in NLU challenges. In 2018, the BERT model emerged as a groundbreaking development in text processing. Its unique transformer-based architecture, which incorporates dual prediction objectives (Masked Language Model and Next Sentence Prediction), along with its utilization of a vast dataset, enabled it to outperform existing methods across various benchmarks. Subsequent models like XLNet [20], and Robustly Optimized BERT Approach (RoBERTa) [21], sought to refine BERT's efficiency. These models introduced subtle enhancements to the original framework and leveraged considerably larger datasets during their training processes. BERT paved the way for the emergence of other sorts of Transformers, such as the renowned GPT [9] and its four generations that followed. The fourth generation GPT-4 [22] is a multimodal large language model created by OpenAI and was released in 2023 to provide human-like text. GPT-4 has the benefit of having a large number of learning parameters and a vast amount of data with which to train. We studied how existing PLMs encode text information produced by a bot or human into high-quality vector representations before beginning the categorization process in this work. Then, using a bot dataset, we fine-tuned these PLMs and evaluated classification performance among PLMs while matching all hyperparameters. The suggested models' performance varies based on several criteria, including the text representation approach utilized.

Over the past ten years, transformer-based models have gained significant traction for their effectiveness across various language-related tasks. However, understanding the mechanisms behind these models often presents a challenge. According to Bolukbasi *et al.*, the obstacle to interpreting

transformer-based models primarily stems from their complexity and the "black box" nature of their mechanisms. Transformer models, such as those used in natural language processing (NLP), rely on deep neural networks with multiple layers and a vast number of parameters. This complexity leads to several specific challenges in interpretation [23]. We chose to focus on bot detection because of its growing importance in the era of expanding social media reach, where distinguishing between genuine and automated interactions is critical.

A. RESEARCH QUESTIONS

The paper aims to address the challenges of bot detection by leveraging the PLMs. The objectives include improving the performance and efficiency of bot identification.

We ask the following research questions that we try to address in this paper:

- *RQ1 :* What are the key linguistic and behavioral features that are most informative to distinguish between human and bot-generated text?
- *RQ2 :* How can transformer-based models, be effectively adapted for bot detection in natural language text?
- *RQ3 :* Does fine-tuning pre-trained language models enhance their performance in bot detection tasks?
- *RQ4 :* How can the predictions of transformer-based models be interpreted to gain insights into the specific linguistic or behavioral cues that led to a classification decision?

B. PROPOSED SOLUTION

This paper proposes an approach to effectively identify tweets on Twitter generated by bots. To encode the text information in a dense numerical vector. A recent text embedding approach based on Transformers was initially used, and the Deep Neural Network (DNN) was trained to classify the vector representations acquired from the preceding PLMs.

The proposed solution involves the following steps:

- Data preprocessing: In the initial stage of data preparation, the tweets are cleaned and tokenized.
- Fine-tuning: Six PLMs were fine-tuned using a bot dataset.
- Text embedding: To produce a feature vector representation of the text, each tweet is processed through a pre-trained LM model, which is encoded using GPT-3 and BERT-based models.
- Training and classification: The model is trained on a labeled dataset containing both bot and human tweets, allowing the patterns and characteristics specific to bot writing to be learned.
- Evaluation and performance analysis: The efficacy of the suggested solution is evaluated using relevant metrics, including accuracy, precision, recall, and the F1-score. We compare the results to demonstrate the effectiveness and efficiency of the PLMs in bot identification.
- Model interpretability: By leveraging SHapley Additive exPlanations (SHAP), BERT Visualizer, and LLM-generated prompting, the decision-making processes of transformer-based language models can be dissected and interpreted. Insights are provided into which features and tokens are most influential in a model's predictions and how attention is distributed across different layers and heads.

C. RESEARCH CONTRIBUTIONS

The main contributions to the state-of-the-art in this field of this study are as follows:

- PLMs are employed to encode an input tweet as a continuous vector space in such a way that words with similar meanings are positioned close to each other in this space, and a DNN is used to identify the input encoding vector as a bot account with a specific probability. The bot detection method focuses on analyzing individual tweets. Then, these PLMs are compared with the application of pre-trained and contextualized embeddings, such as GloVe and Word2Vec, as inputs for a BiLSTM in the classification task.
- The approach employs an advanced text embedding technique based on a Pre-trained Language Model. It effectively identifies similarities among words within a tweet, subsequently translating these into a refined numerical representation. The suggested architecture allows us to achieve cutting-edge performance in bot detection. (roughly 93% F1-score).
- We contribute to the field by demonstrating how SHAP and PLM-based prompting can be used to interpret the complex decision-making processes of pre-trained language models.

D. ORGANISATION OF THIS PAPER

The remainder of this paper is organized as follows. In Section II, we discuss previous work in the area of detecting bots in Online Social Networks. In Section III, this paper reviews the current research literature on techniques for distinguishing between authentic and computer-generated text. Following this, Section IV outlines the setup of our methodology. It elaborates on our approach to bot detection, encompassing the datasets employed for fine-tuning and training the Pre-trained Language Models (PLMs), along with the methods used for assessing their performance. The study provides experimental findings in Section V. Finally, Section VIII is dedicated to the conclusion with ideas for future work.

II. RELATED WORK

The prevalence of social bots has raised significant concerns regarding information integrity and social dynamics online. This surge necessitates advanced detection methods, where transformer-based models have emerged as potent tools. The ability of these models to understand and process natural language nuances offers a promising avenue for distinguishing between human and bot-generated content. This literature review delves into how these fine-tuning transformer models, particularly in the context of social bot detection, can significantly enhance their efficacy and accuracy.

Bots, including social bots and Sybil accounts, have been implicated in contaminating social media discussions across various contexts. Instances of online manipulation facilitated by bots encompass political discourse and extend to other domains as well [24], fake news [25], [26], Bot-IoT [27], [28] and public health [29] are among the topics covered. It is important to mention that in certain rare cases, bots have been deployed to offer helpful interventions, rather than for criminal goals [30]. Ellaky *et al.*'s recent systematic review [31] introduced a classification of three distinct methods for differentiating between bots and humans. These methods encompass: (a) graph-based techniques, (b) approaches relying on crowd-sourcing and human computation, and (c) learned-based models designed to distinguish bots from human users. The framework employed in this study falls under the third category. A recent survey proposed a categorization scheme that highlights the wide range of variability and diversity exhibited by bots regarding their conduct, abilities, and intentions. This scheme emphasizes the different characteristics and classifications observed among bots [3].

In contrast to static word embedding methods like Skip-Gram and Continuous Bag of Words, as mentioned in [32], language models can understand the context surrounding words. Consequently, they can assign different values to words based on the context in which they appear. The bidirectional encoder representations from transformers (BERT) language model is one of the most prominent, and it has been shown to perform well in text classification tasks. Most methods created in the NLP field now employ PLMs as their basis. These models can encode broad language information useful for downstream tasks by utilizing huge corpora during a pretraining phase. These models' modeling capabilities have been further enhanced with the introduction of PLMs based on the transformer architecture. As proven in [33], bot detection methods are frequently insufficiently strong to extend applicability to social bot scenarios beyond those included within the training dataset.

Various research articles have addressed the subject of bot detection and classification. These works have explored a range of methodologies, including LSTM, deep learning techniques [34], Hidden Markov models [35], and contemporary pre-trained language models [19], [36], and other approaches were employed in these studies. The literature review plays a vital role in the research process as it allows researchers to explore the current research topics related to bot detection techniques in online social networks (OSNs). It helps researchers identify the strengths and weaknesses of existing approaches, as well as uncover new trends and directions for further investigation. By conducting a literature review, researchers gain valuable insights into the current state of research, identify areas that need improvement, and propose innovative approaches to enhance bot detection in OSNs. In this section, we attempt to summarize the important works done in this study area.

Kudugunta *et al.* [37] provided research that deals with bot tweet identification based on content and metadata. To represent Twitter content, the authors combined an LSTM layer with GloVe embeddings [7].

Research on automatic bot identification from individual tweets posted by accounts not included in the training set was presented by the authors in [38]. They tested their method by dividing the dataset into training and test subsets with no common users. They used RoBERTa to develop the bot tweet identification technique and examined accuracy problems arising while detecting GPT-2 based tweets. Their findings indicated that the dataset used to train the model significantly impacted the proposed classifier's generalization capacity.

Harald and Johansso in [39] explored the potential risks of malicious actors using modern neural language models to produce fake content that appears to be written by real people and evaluated various detection algorithms and their effectiveness in identifying texts generated by language model-based generators. They also compared the performance of different models using data from in-distribution, out-of-distribution, and in-the-wild neural language models managed in various ways.

Pu *et al.* [40], primarily concentrated on two key areas of research. Firstly, they aimed to enhance the understanding and effectiveness of defenses in real-world scenarios. They investigated the performance of existing defense mechanisms and explored ways to improve their robustness against attacks. Secondly, they focused on understanding and enhancing the performance of defenses against adaptive attackers. Examining evasion strategies that were practical and cost-effective, they specifically focused on those that did not require any queries to the defender's model.

MARTÍN-GUTIÉRREZ *et al.* [41] presented a multilingual methodology, leveraging Deep Learning techniques, to assist users in evaluating Twitter account credibility. To generate text-based user account features, their approach integrated state-of-the-art Multilingual Language Models, which are subsequently fused with metadata to construct an input vector for a Dense Network, referred to as Bot-DenseNet.

In the study conducted by Dukic *et al.* [42], the authors employed BERT contextualized embeddings to build Logistic Regression and Deep Neural Network models for bot and gender prediction.

Shubham Kumar *et al.*, in their study [43], presented an approach for detecting bots on Twitter, using a set of neural networks. This set incorporated a Text CNN and an LSTM model, both augmented with BERT embeddings. Additionally, the study addressed the problem of data imbalance in Twitter datasets through the use of the Language Model Oversampling Technique. This technique involved creating a bot language model that mimicked the patterns and characteristics of bot tweets. Guo *et al.* [44] integrated the pre-trained language model BERT with Graph Convolutional Networks (GCNs) to identify social bots.

This research proposed by Liu *et al.* [45] aimed to identify and classify social bots on Weibo, a popular Chinese microblogging platform. The methodology involved topic expansion, opinion sentence recognition, and transfer learning to

identify and classify social bots into three categories: polluters, commenters, and spreaders. They also addressed the problem of data scarcity, which was caused by the wide gap between the microblog text length and the topics, as well as the restricted quantity of data available from social bots.

In their research, Heidari *et al.* [46] introduced a model that employed BERT to classify sentiments in tweets. The objective was to discern topic-agnostic attributes for use in a model designed to detect bots on social media. By applying BERT to analyze the sentiment of tweets, the model extracted valuable insights that contributed to the detection of social media bots.

Another study, Wu *et al.* [47] presented the RGA model, which integrated ResNet, BiGRU, and an Attention mechanism, to detect social bots on the Sina Weibo platform. This approach was structured into four key components: gathering and categorizing data, extracting features, employing active learning, and carrying out bot detection. The research employed a dataset containing 20,000 classified user profiles, which they analyzed using advanced deep neural network techniques. Additionally, they used the active learning aspect to increase the size of the dataset to 300,000 users.

The work by Loukas and Ioanna [48] stood out for its approach using deep learning techniques. Their study presented two distinct methods to identify bots on Twitter. The first method focused on a comprehensive feature extraction process to differentiate between bot and human accounts. The second method employed a sophisticated deep learning architecture that combined an attention mechanism with a BiLSTM layer, specifically designed for classifying tweets.

In their study, Zeng *et al.* [49] developed a social media bot detection framework called MRLBot. This system combined two distinct models: the DDTCN, which used a Transformer and CNN encoder-decoder to analyze user behavior, and the IB2V, which focused on mapping relationship networks through random walks in community contexts.

Shangbin Feng *et al.* [50] introduced a bot detection framework for Twitter in their study. This architecture utilized the topological configuration of heterogeneous graphs formed by users and captured the varying degrees of influence among them. A heterogeneous information network was established, featuring users as nodes and diverse relationships as edges. Subsequently, relational graph transformers were applied to depict the heterogeneous influence among users and to derive representations of nodes. Furthermore, semantic attention networks facilitated the aggregation of messages across users and relationships, enhancing the detection of Twitter bots with heterogeneity-aware.

A Previous endeavor [51], employing Convolutional Neural Networks (CNN) and BiLSTM for pre-training purposes, are comparatively traditional. This research distinguishes itself by integrating avant-garde models such as BERT and GPT, and then applying fine-tuning, thus representing a considerable progression in the field. This study by the Garcia-Silva *et al.*demonstrated that fine-tuning pre-trained language models for bot detection on social media platforms, especially Twitter, led to enhanced classification performance compared to solely using pre-trained embeddings. The results underscored the effectiveness of language models such as Open AI GPT and BERT.

BOTTRINET, a unified embedding framework developed by Jun Wu *et al.*for social bot detection, leveraged textual content to profile accounts and detect bots. Through its Embedding Network, Triplet Loss function, and Triplet Selector module, BOTTRINET refined raw content embeddings using metric learning techniques. These techniques aimed to maximize the distance between bot and genuine user embeddings.

Various performance metrics have been employed to evaluate the methods discussed in the literature review. The specific metrics and their corresponding values can be found in Table 1.

A critical gap identified in the existing literature is the absence of bot-specific data in the pre-training phase of models used for bot detection. Traditionally, PLMs are developed using vast, general datasets that do not specifically include or focus on bot-generated content. This lack of bot-oriented data during pre-training could limit the models' ability to effectively discern the nuanced differences between human and bot-generated text. In this study, we address this gap by fine-tuning PLMs with a specialized dataset that includes a substantial proportion of bot-generated content. This approach aims to enhance the model's sensitivity to the subtleties of bot-like patterns and behaviors, thereby improving its accuracy in detecting social bots.

III. BACKGROUND

In this study, we delve into the foundational concepts and technologies that form the basis of the bot detection. This includes an exploration of Pretrained Language Modeling and Transfer Learning, where we discuss how PLMs serve as a starting point for learning specific tasks. We then focus on BERT-based Language Models, examining their architecture and effectiveness in understanding the context of text. The subsection on GPT (Generative Pre-trained Transformer) covers its generative capabilities and applications in natural language understanding. Subsequently, we discuss the Feedforward Neural Network (FNN), highlighting its role in processing input data and its integration with advanced language models in bot detection tasks. Finally, we integrate interpretability tools such as SHAP and BERT Visualizer into the analysis of PLMs for bot detection significantly enhancing our understanding of these models.

A. PRETRAINED LANGUAGE MODELING AND TRANSFER LEARNING

Transfer learning is a deep learning technique that aims to transfer learned knowledge from one domain to a new target domain. This strategy is primarily employed to overcome the challenge of limited training datasets, which often leads to overfitting and subsequently affects the performance of the model [71]. Traditional machine learning methods typically focus on single-task learning in isolation. Conversely, transfer

TABLE 1. A comparative analysis of the mentioned relevant works.

Reference	Applied Techniques	Performance classification	Dataset
Kudugunta ***et al.*** [37]	Contextual LSTM (200D GloVE) by combining text content and metadata	F1-score: 96%	The dataset consists of a total of 8,386 user accounts and more than 11,834,866 tweets, which is a mixture of genuine accounts and social spambots [52].
Tourille ***et al.*** [38]	The classification score is generated by feeding the concatenated RoBERTa (small version) into a two-layer feedforward neural network	F1-score: 91.3%	GPT-2 to produce a new set of tweets combined with the dataset from Fagni et al. [53].
Harald Stiff and Fredrik Johansson [39]	RoBERTa,GPT-2, GROVER	F1-score: 84%	news articles and social media texts, including both genuine human-written texts as well as texts generated by language models controlled with GeDi and PPLM.
Pu ***et al.*** [40]	RoBERTa-Defense	F1-score: 86.3%	Dataset (9000 samples) used for training and testing defense model including synthetic data produced by GROVER [54] and RealNews.
MARTÍN-GUTIÉRREZ ***et al.*** [41]	RoBERTa + metadata	F1-score: 77%	The study uses several public datasets : [55], [56], [52], [57].
David Dukić ***et al.*** [42]	BERT + metadata	Weighted F1-score: 83.35%	The cleaned dataset consists of 288,178 tweets for training and 190,211 tweets for testing. The dataset includes English tweets from the PAN competition data set [58].
Shubham Kumar ***et al.*** [43]	BERT with CNN & LSTM-Attention	F1-score: 97%	The dataset used in the study comprises a total of 8.4 million user tweets and 4.8 million bot tweets [52].
Guo ***et al.*** [44]	BERT with GCN	F1-score: 81%	The study uses social bot detection benchmarks : [59], [55], [60], [61], [62], [56].
Lie ***et al.*** [45]	Sentence-BERT with TextCNN & Transfer learning	F1-score: 92.3%	The dataset is collected by writing crawler code, and the second part is obtained from the Weibo API. It consists of 8,000 Weibo accounts for training and 2,000 accounts for testing. The dataset is labeled with three categories: polluters, commenters, and spreaders.
Heidari and Jones [46]	BERT & FFNN	F1-score: 94.7%	Cresci 2017 dataset [52].
Wu ***et al.*** [47]	RGA (ResNet, BiGRU, and Attention mechanism)	F1-score: 98.86%	30 features including metadata-based, content-based, interaction-based, and timing-based features. The data collection is performed using a web crawler to collect user data from Sina Weibo and manually label it (150,000 social bots and 150,000 normal users) [52].
Loukas Ilias and Ioanna Roussaki [48]	BiLSTM layers with an attention mechanism.	F1-score: 86.43%	The dataset used includes two publicly available Twitter datasets: the Cresci-2017 dataset [52] and the Social Honeypot Dataset [63]. The Cresci-2017 dataset consists of 3474 genuine accounts and 4912 social spambots, while the Social Honeypot Dataset contains 19276 legitimate users and 22223 content polluters.
Zeng ***et al.*** [49]	A transformer combined with a CNN encoder-decoder.	F1-score: 97.85%	The study uses social bot detection datasets: Cresci-2015 [64] ,Social-Spammer [65] ,TwiBot-22 [66],MicroblogPCU [67].
Shangbin Feng ***et al.*** [50]	graph-based & heterogeneity-aware	F1-score: 88.21%	TwiBot-20 comprises a dataset of 229,573 Twitter users, encompassing 33,488,192 tweets, 8,723,736 items related to user properties, and 455,958 follower connections [68].
Garcia-Silva ***et al.*** [51]	BiLSTM + dynamic and ELMo embeddings. CNN + dynamic and ELMo embeddings GPT	F1-score: 0.8074% F1-score: 80.73% F1-score: 85.33%	The dataset contained 500,000 tweets in total. Of these, 279,495 tweets originated from 1,208 human accounts, while 220,505 tweets were posted by 722 bot accounts [69].
Jun Wu ***et al.*** [70]	Metric learning techniques	F1-score: 94.36%	Cresci-2017, consisting of three bot account categories and five bot sample sets [52].

learning boosts performance through the utilization of shared parameters from a model that has been pre-trained on data relevant to analogous tasks. This method allows the model to utilize knowledge gained in previous training, thereby improving accuracy, decreasing the need for large training datasets, and accelerating learning compared to conventional machine learning techniques.

Initially prominent in the field of image classification, the principle of transfer learning has been effectively demonstrated using pre-trained network models like VGG [72], and ResNet [73]. In the Natural Language Processing (NLP), the emergence of PLMs such as GPT [9], ELMo [74], and BERT since 2018 has significantly enhanced the efficacy of various NLP tasks. Notably, models like BERT and ELMo have addressed a major shortcoming of earlier language models such as GPT, which depended on a unidirectional learning approach. This approach often neglected the crucial element of 'context' in language, which is essential for many NLP tasks. As a result, the potency of pre-trained representations was somewhat constrained. Addressing this, BERT and its variations have introduced more effective, contextually rich bidirectional learning methodologies. By fine-tuning all previously learned parameters, BERT can be used for downstream tasks. The availability of open-source code, improvements in hardware accelerators, and continuous research on fine-tuning existing language models have made it easier for various actors to train and fine-tune models specifically for their NLP objectives.

The Transformer network [75] addresses these shortcomings. The parallelization of a long-range relationship is ensured by the encoder/decoder layers and self-attention in the transformer network. As previously said, labeled data, which is limited in quantity, serves as the foundation of deep learning models. In deep learning applications, supervised learning is the predominant approach, requiring human-annotated instances to facilitate model learning. However, transformer networks offer an alternative avenue through self-supervised learning known as pseudo-supervision, enabling training from unlabeled datasets. Meanwhile, there is a lot of unlabeled data available. A supervised learning application has the benefit of producing models that perform exceptionally well on certain datasets. Creating human-annotated labels is a time-consuming procedure that necessitates using a domain expert, who is in short supply. Supervised learning models face difficulties in generalization and are prone to spurious correlations. This is because they heavily rely on known training patterns and struggle to handle unknown samples effectively.

B. BERT-BASED LANGUAGE MODELS

BERT, a novel pre-trained language model, has significantly advanced the performance in various NLP tasks. Unlike traditional models, BERT generates contextual embeddings. These are fixed-size vector representations that vary not just based on the words themselves but also the surrounding context. For illustration, the word 'bank' would be represented differently depending on its association with 'economy' or 'river', leading to distinct contextual vectors. Consequently, these context-specific representations can be combined to form a contextualized fixed-size vector for a text. This could be achieved, for example, by averaging the vectors of all words in a particular text. During the pre-training phase, BERT learns contextual representations of words by training on a massive unlabeled text corpus. As training objectives, the proposed approach incorporates Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, a portion of the input words are randomly masked, and the model must recognize them based on their context. NSP relies heavily on predicting the co-occurrence of two sentences. BERT can get contextual information and deep word representations from these pre-training operations. BERT may be customized for usage in several NLP applications once it has been pre-trained. BERT-base and BERT-large models differ primarily in their architectural configurations. The BERT-base model is composed of 12 transformer layers, features 768 hidden layers, includes 12 attention heads, and encompasses roughly 136 million parameters. In contrast, the BERT-large variant is more complex, comprising 24 transformer layers, 1024 hidden layers, 16 attention heads, and an estimated 340 million parameters. The majority of the BERT model versions were created by modifying the last layers to adapt to a range of NLP situations. BERT's architecture, which includes a multilayer stack of transformer encoders and a self-attention mechanism, has proven to be highly effective in capturing intricate contextual representations of text. This has resulted in significant advancements in various NLP tasks. In this investigation, three BERT variants were employed, DistilBERT [76], XLM-RoBERTa [77], and RoBERTa [21].

1) **RoBERTa** model implements various enhancements to the BERT architecture, including the utilization of an expanded dataset and increased batch size for training, the elimination of the next-sentence prediction task, the extension of sequence lengths during training, and the dynamic adjustment of the masking pattern used on the training data. Consequently, RoBERTa has demonstrated superior performance over BERT in natural language processing (NLP) benchmarks, such as the General Language Understanding Evaluation (GLUE) [78], the Stanford Question Answering Dataset (SQuAD) [79], and the RACE dataset [80]. BERT and RoBERTa, commonly constrained by a maximum input sequence length of 512 tokens (context window), necessitate adaptations for the processing of extended texts within transformer-based models. One approach to address this limitation involves truncating the input sequence to the initial 512 tokens;
2) **XLM-RoBERTa** is a pre-trained language model (PLM) transformer that is frequently employed in multilingual tasks. It undergoes pretraining as a masked language model on a vast corpus of data, consisting of 100 languages and approximately 2.5 TB of re-

fined common crawl data, XLM-RoBERTa is purposefully crafted as a multilingual version of RoBERTa. The acronym "XLM" denotes its capabilities in cross-lingual language modeling;

3) **DistilBERT** is a compact and efficient Transformer model that is trained by distilling the knowledge from the larger BERT base model. It has approximately 40% fewer parameters compared to the *bert-base-uncased* model. Despite its smaller size, DistilBERT maintains over 95% of BERT's efficacy on the GLUE language understanding benchmark. In addition, DistilBERT offers a significant speed improvement, running approximately 60% faster than BERT while maintaining comparable performance;

C. GPT (GENERATIVE PRE-TRAINED TRANSFORMER)

GPT [9] is a cutting-edge class of natural language processing models that combines pre-training on a large corpus of text data with fine-tuning for specific language tasks. Developed by OpenAI, GPT models are based on a Transformer architecture. They have shown impressive ability in a variety of natural language understanding and generating tasks, including summarization, question-answering, text completion, translation, and chatbot applications. GPT models employ unsupervised learning during pre-training, allowing them to capture and learn intricate linguistic patterns, contextual relationships, and semantic nuances from diverse textual sources. This pre-trained knowledge is subsequently fine-tuned on task-specific labeled data, enabling GPT models to adapt to specific language tasks effectively. GPT models have achieved state-of-the-art performance in numerous natural language processing benchmarks and have significantly advanced the field of artificial intelligence and language understanding.

In their 2022 publication, OpenAI introduced an innovative text embedding model, an extension of their GPT-3 technology. This model's primary goal is to generate superior vector representations for textual data. Its embedding technique is notable for its ability to grasp and reflect the semantic similarities in segments of text. The detailed process of this embedding is illustrated in Fig. 1. In terms of input processing, the model begins by appending two distinct tokens to the input text, namely [SOS] at the start and [EOS] at the end. Subsequently, this modified input sequence is transformed by a Transform Encoder E, resulting in a dense vector representation denoted as V_x. The crucial step involves extracting the hidden state from the model's final layer, which correlates to the input text. Specifically, the embedding is derived from the hidden state associated with the [EOS] token in the last layer.

The effectiveness of GPT-3 and its embedding model has motivated us to incorporate them into our approach for representing input tweets. This inclusion aims to enhance the accuracy of classification tasks. To obtain an embedding for a text string, we send the text along with the desired embedding model ID to the embeddings API endpoint. For example, we specify the embedding model ID as "text-embedding-ada-002" to indicate the specific embedding model.

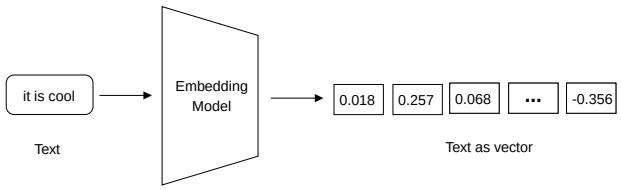


FIGURE 1. Text representation using the embedding model

D. FEEDFORWARD NEURAL NETWORK (FNN)

This study opted for the feedforward neural network model [81], for the concluding stage of the newly developed bot detection model. A two-layer feedforward neural network makes up the head. We employ 0.1 as the dropout rate on both the input and hidden layers, and ReLU as the activation function. Binary Cross Entropy loss is minimized during the training of the classification model. We train the transformer model for 5 epochs using a learning rate of $2e^{-5}$, while the classification head is trained with a learning rate of $1e^{-3}$. We employ a mini-batch size of 16 and Adam as the optimizer. To address the issue of overfitting in neural networks, a commonly employed technique called Dropout has been incorporated. Dropout, initially introduced in [82], aims to mitigate overfitting by randomly deactivating neurons during each epoch with a specified probability. This technique helps prevent the network from relying too heavily on specific neurons and encourages more robust generalization to unseen samples.

E. EXPLAINABILITY OF PRETRAINED LANGUAGE MODELS

A frequently discussed ethical concern in today's society is the problem of transparency. It is crucial to have the capability to elucidate an artificial intelligence's prediction or decision to a customer or an ordinary AI user [83]. Additionally, explaining a sophisticated learning algorithm not only aids in understanding the model's decision-making process but also facilitates the optimization of the model by pinpointing the critical tokens necessary for identifying a bot tweet.

1) SHapley Additive exPlanations (SHAP)

The **SHAP** technique, introduced by Lundberg and Lee [84], offers post-hoc explanations across various model types regarding the contribution of individual variables. Determining the Shapley values for features within a specific instance involves simulating assorted configurations of feature values, including scenarios where a feature might be entirely missing. For every configuration, the deviation between the model's predicted value and the average prediction across actual data is computed. The formal definition of a feature's Shapley value, denoted as φ_i, is as follows (see Equation (1)).

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} (f_x(S \cup i) - f_x(S)), \quad (1)$$

M denotes the number of features, S represents the set of features, and f_x signifies the prediction function at a given time x, with $f_x(S) = E[f(x)|x_S]$ illustrating the expected value of $f(x)$ conditioned on x_S. Here, i corresponds to the ith feature in the dataset.

The SHAP approach is additive, whereby a prediction is articulated as the aggregate of the individual contributions of the variables (denoted by the Shap value φ_i) in conjunction with the base value φ_0. The base value is established as the average prediction for the entire dataset, as delineated in Equation (2).

$$f(x) = y_{pred} = \varphi_0 + \sum_{i=1}^{M} \varphi_i z_i'. \tag{2}$$

with, y_{pred} the predicted value of the model for this example, $z' \in \{0,1\}^M$ when the variable is observed $z_i' = 1$ or unknown $z_i' = 0$.

IV. PROPOSED MODEL

In this section, we will comprehensively describe the model proposed in this work. The primary goal is to investigate the possibility of automatically detecting bot activity from individual tweets made by accounts that are not included in the training set. This involves a binary classification task to determine whether a human or a bot generates each tweet. We remove the duplicate rows after the preprocessing. Then, the dataset is divided into train and test subsets so that there are no common users. The workflow for this research is described in Fig. 2. First, this work was started by cleaning the input text and converting it into a numeric format using an embedding process based on the PLM. The embedding result was then used to train a Deep Neural Network (DNN) for a classification task. The output layer made its final decision regarding the category of the input text.

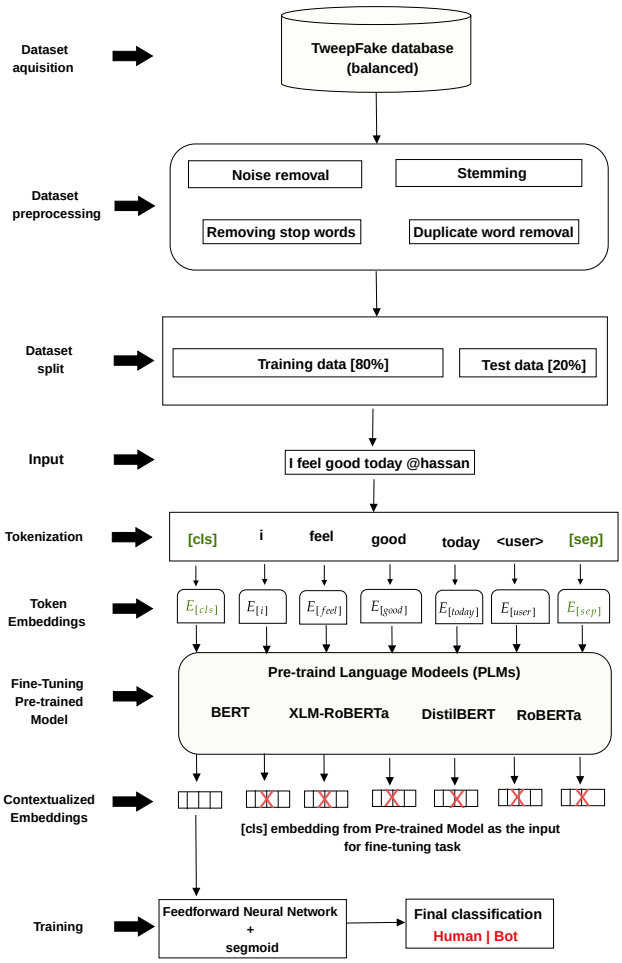


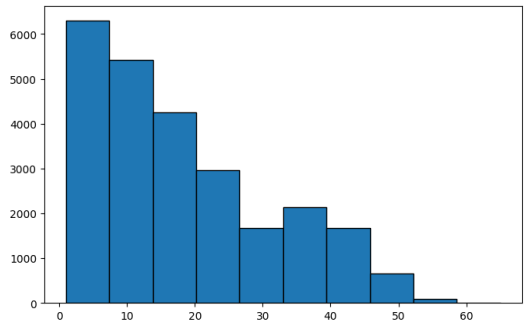
FIGURE 2. Architecture of the bot detection model

A. DATASET AND FEATURES

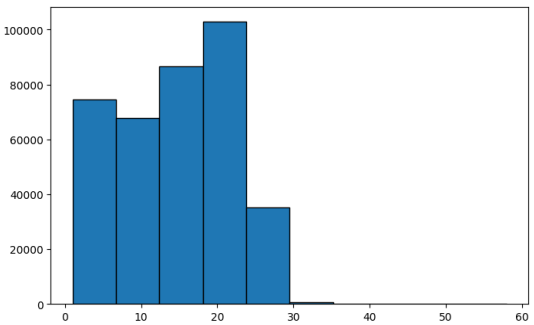
The first dataset used in this research is called *TweepFake* and contains $25,572$ tweets, half of which are generated by humans and the other half by bots. The fake tweets are generated by various generation techniques, i.e., Markov Chains [85], RNN [86], RNN+Markov, LSTM [87], GPT-2 [88]. This dataset was gathered by Fagni ***et al.*** [53] and has been made public to assist researchers in testing their techniques for Twitter bot detection. The second dataset utilized in this study comprises 1,140 bot accounts and 1,140 human accounts, forming a benchmark dataset referred to as the "fox8-23", which can be accessed publicly at github.com/osome-iu/AIBot_fox8. In total, the dataset contains 218,245 human accounts and 149,783 bot accounts. The datasets are split into train and test sets, with no common users between them. Fig. 3 gives an overview of the variability of text sequence length in the TweepFake and fox8-23 datasets. As we can see in Fig 3a, the plot has a longer tail on the left side, indicating that most tweets are very short. Therefore, we have set the maximum sequence length to 60 for fine-tuning the pre-trained models. In contrast, the fox8-23 dataset exhibits a more spread-out distribution, with a noticeable peak around the 20-token mark. This dataset also shows a substantial number of tweets with lengths up to 30 tokens, followed by a steep decline for tweets longer than 30 tokens. These differences highlight the variability in tweet lengths across different datasets, which can impact the performance and training of models for bot detection.

Table 2 provides a visual representation of selected samples from the datasets used for bot detection, showcasing instances of both bot-generated and human-generated tweets.

We applied Latent Dirichlet Allocation (LDA) to identify key topics prevalent in the bot-generated content. Fig. 4 illustrates the top words for three identified topics. *Topic 1* highlights common words like "<user>", "<url>", and "to", indicating general patterns in bot communication. *Topic 2* shows the prominence of "rt" and "<hashtag>", suggesting frequent use of retweets and hashtags in bot activity. *Topic 3* also emphasizes "rt" and "<user>", further underscoring the repetitive nature of bot interactions. These visualizations



(a) TweepFake



(b) fox8-23.

FIGURE 3. Distribution of sequence lengths of tweets

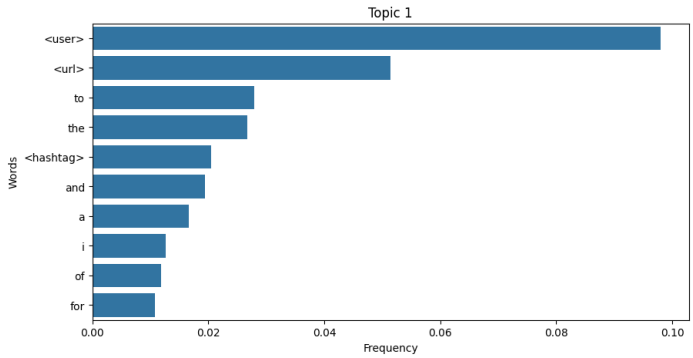
TABLE 2. Example subset of data

Tweet example	Class Type
justin timberlake really one of the goats if you think about it	Human
Respects on the Upt of the I good with the people of West Bengal	Bot
This is one of those tweets that will make you want to go WOW!!	Bot
Earlier today, I spoke with Premier @jkenney about the latest developments on the flooding in Fort McMurray and assured him the federal government stands ready to help. More on our call here: https://t.co/q53RJk7oM7	Human

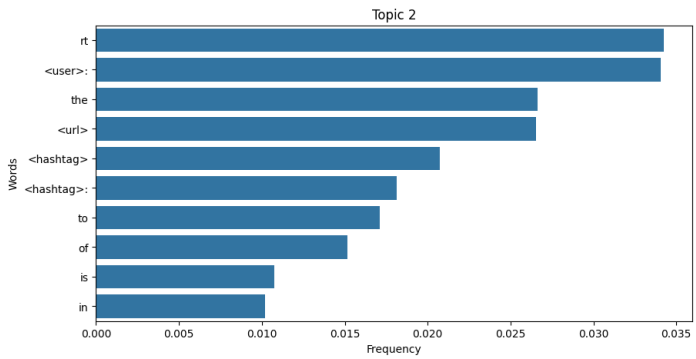
provide an understanding of the linguistic characteristics and strategies employed by bots in social media. Additionally, Fig. 5 illustrates the t-SNE inter-topic distance map, which effectively visualizes the distribution and relationships between various topics within the tweet dataset. The map demonstrates well-separated topics, indicating that the LDA model has successfully identified distinct thematic clusters. This visualization is instrumental in analyzing the characteristics of bot-generated tweets.

B. TEXT CLEANING AND PREPROCESSING DATA

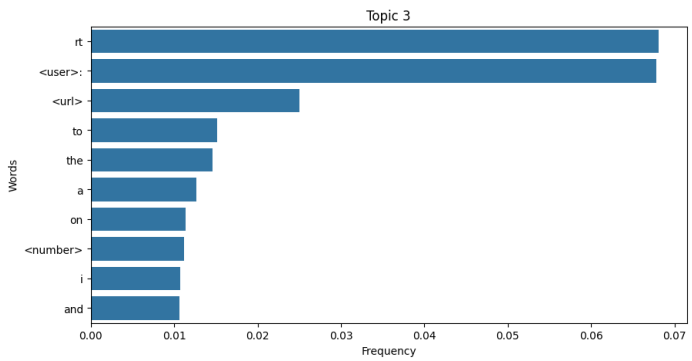
The tweets posted by OSN users are represented by the content feature. These tweets are written unusually, including text, hashtags, numbers, emojis, and so on. Text data must be cleaned and preprocessed using Natural Language Processing (NLP) methods. Preprocessing text data assists in training a model with relevant data to maintain the pattern



(a) Top words in Topic 1 from bot-generated tweets, highlighting common user mentions and URLs.



(b) Top words in Topic 2 from bot-generated tweets, showing the prevalence of retweets and hashtags.



(c) Top words in Topic 3 from bot-generated tweets, emphasizing the use of retweets and user mentions.

FIGURE 4. Topic Analysis of Bot-Generated Tweets Using Latent Dirichlet Allocation (LDA) on the combined datasets.

necessary to develop reliable DL models. Prior to initiating data processing, it is imperative to undertake text cleaning. This process involves the removal of superfluous elements present in tweets. Specifically, this includes the elimination of punctuation, non-essential symbols, excess spaces, and line breaks. Such a step ensures that the data is primed for subsequent analysis, free from irrelevant textual noise. This process is effective for improving text comprehension by deep learning models. Before applying word embedding, we preprocess the tweets by tokenizing them using the tokenization methods provided by the creators of PLMs. This ensures that

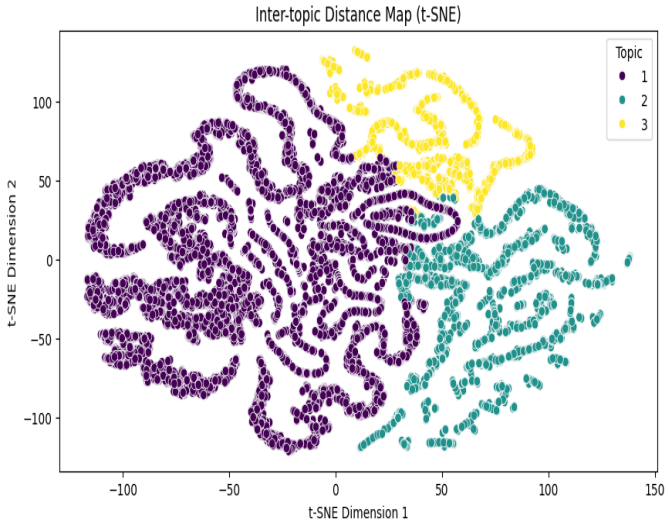


FIGURE 5. t-SNE-based 2D visualization plot

the tweets are appropriately prepared for further analysis and embedding.

- Tokenization involves the process of decomposing a sentence into its individual components, usually using a tokenizer like WordPiece [89] or SentencePiece [90]. This process entails dividing the sentence into its fundamental elements referred to as tokens.
- We replace any instances of numbers, user mentions, hashtags, and URLs with specific tags. Hashtags are replaced with the tag $< hashtag >$, URLs are replaced with $< url >$, numbers are replaced with $< number >$, and user mentions are replaced with $< user >$. This preprocessing step helps standardize and anonymize these elements in the text data.
- Similarly, all emojis are replaced with $< smile >$, $< heart >$, $< lolface >$, $< neutralface >$ or $< angryface >$, depending on the specific emoji.
- All tokens are converted to lowercase.

C. FINE TUNING ON DOWNSTREAM TASK

For language processing tasks, top models like BERT, RoBERTa, DistilBERT, and XLM-RoBERTa used pretraining a language model from scratch. The data-driven training process leverages self-supervised learning on a larger unlabeled dataset, capitalizing on the computational capabilities of high-processing devices like graphics processing units (GPUs). Pretraining from scratch is computationally challenging and costly, making the utilization of powerful computing resources essential for managing the training process effectively. In this methodology, we engaged multiple pre-trained language models, applying a fine-tuning process on the specific dataset. This procedure entailed adapting an already trained language model, integrating its established architecture with an additional dense classification layer at the end. We then trained the model on specific dataset, typically with a small number of epochs. The process of fine-tuning allowed for the tailored adjustment and modification of the network weights inherent in the original language model, specifically aligning them with the unique application requirements. Consequently, this adaptation enhanced the model's encoding efficiency, optimizing its performance for the targeted classification challenge.

Increasing domain knowledge in bot detection can be accomplished by fine-tuning a particular dataset. For example, if we wanted to increase the model's capacity to separate human tweets from bot tweets in social media, we could include more information about social media platforms. With the aid of this information, the PLMs will be better able to comprehend the linguistic conventions and terminologies used there. In the research literature, various detection models have been proposed to distinguish between texts generated by humans and those generated by bots. These models are discussed in more detail in Section II of the paper, the experiments in this study specifically focused on Transformer-based detection models. This choice was motivated by the promising results demonstrated by Transformer-based models in previous research. In this study, we fine-tune the PLMs on *TweepFake* and *fox8-23* datasets. We think that by using the knowledge from the original model during training, the performance of the classifier might be enhanced. The [CLS] token added at the beginning of each sentence or document is used to derive the predicted outcomes.

In this research, we evaluated various language models: RoBERTa, DistilBERT, XLM-RoBERTa, and both the base and large variants of BERT, along with GPT-3. Each of these models employs the transformer architecture, renowned for delivering top-tier results in numerous text processing benchmarks.

D. TEXT EMBEDDINGS

Text embedding refers to the process of converting textual data into a meaningful vector representation. It involves encoding the text into numerical values that capture the semantic and contextual information of the text. This encoded representation enables the text to be effectively processed and analyzed by machine learning models. Various text representation techniques have been explored in literature, including TF-IDF [91], [92]. Word embedding models are also prominent, with Word2Vec highlighted in Baek ***et al.***'s study [93], GloVe as detailed by Pennington ***et al.*** [7], and fastText, as mentioned in Taher ***et al.*** citetaher2022automatic. Additionally, recent advancements involve pre-trained Transformer models like BERT, as explored by Devlin ***et al.*** [19], and GPT. Various strategies were researched and compared since the Transformer-based text embedding method was required. Moreover, the creators of GPT-3 showcased in their latest research, published in January 2022, the effectiveness of their text embedding methodology. This approach yielded superior-quality vector representations of textual data and set new benchmarks in linear-probe classification. Their findings indicated a significant improvement over the performance

TABLE 3. Confusion Matrix

	Predicted: Bot	Predicted: Human
Actual: Bot	True Positive (TP)	False Negative (FN)
Actual: Human	False Positive (FP)	True Negative (TN)

metrics of the previously leading text embedding models. The remarkable results and advancements demonstrated by the GPT-3 text embedding approach prompted us to integrate this model into the proposed approach. Pre-trained embeddings are utilized to train the LSTM classifier instead of starting from zero. These embeddings are derived from Twitter to capture the platform's specific linguistic traits, and Google News to address the casual vocabulary frequently found on the social network.

- glove.twitter200d: 200 dimension embeddings generated from Twitter (27B tokens, 1.2M vocabulary) using GloVe [7].
- word2vec.GoogleNews-vectors-negative300: model, which has a 300-dimensional vector space, is based on the Word2Vec technique. This method was developed by Tomas Mikolov and his team at Google [94].

E. CLASSIFICATION MODEL

After obtaining the text embedding, it is passed through two ReLU-activated layers. The first layer has a size of 256, followed by a second layer with a size of 32. The resulting output offers the prediction of the class associated with the input message: C0 (human) or C1 (bot).

V. MODEL PERFORMANCE

In this section, we present the results of the experiments conducted to evaluate the model proposed in the current paper. These experiments contained an FNN on top of the fine-tuned PLMs tested with six types of text representation: BERT (base and large), RoBERTa, DistilBERT, XLM-RoBERTa, and GPT-3-based embedding.

A. PERFORMANCE EVALUATION

In evaluating the efficacy of each method, diverse metrics are utilized. These include precision, recall, F1-score, accuracy, and the Area Under the Receiver Operating Characteristic Curve (AUC/ROC).

1) Confusion matrix

To accurately examine the effectiveness of a detection model, the initial step involves constructing a confusion matrix that quantifies two types of errors: false positives (incorrect alerts triggered for human-generated content) and false negatives (bot-generated tweets that go undetected). Table 3 describes the components of a confusion matrix.

True real label = predicted label
False real label $\neq$ predicted label
Positive prediction is bot
Negative prediction is human.

The following evaluation measures are then discussed in detail:

1) **Accuracy**
It measures the proportion of correctly classified examples out of the total number of examples in the dataset. Accuracy is a widely used performance metric to assess the overall effectiveness of a classifier:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}$$

2) **Precision**

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) **Recall**

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) **F1-Score**
The F1-Score is defined as the harmonic mean of recall and precision and provides a balanced measure of both metrics. It is calculated using the following formula:

$$F1 - Score = 2 \times \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})}$$

2) Receiver Operating Characteristic (ROC Curves)

Classification models that yield a probabilistic score within the range of $[0, 1]$ instead of discrete classes $\{0, 1\}$ can be optimized by selecting a threshold that differs from the default value of 1/2. The Receiver Operating Characteristic (ROC) curve is a graphical representation used to evaluate the performance of binary classification models. It plots the true positive rate (TPR) or sensitivity against the false positive rate (FPR) at various threshold settings. The TPR is calculated as the number of correct positive predictions divided by the total actual positives, while the FPR is the number of incorrect positive predictions divided by the total actual negatives. The ROC curve illustrates the trade-off between sensitivity (the ability of the model to correctly identify positive instances) and specificity (the ability of the model to correctly identify negative instances) across different thresholds. It helps in assessing the diagnostic ability of the classifier.

The True positive rate (TPR) is equivalent to the recall metric. It represents the proportion of correctly identified positive instances out of all actual positive instances. Mathematically, it is defined as follows:

$$TPR = \text{Sensitivity} = \frac{TP}{TP + FN}$$

False positive rate (FPR) is defined as follows:

$$FPR = 1 - \text{Specificity} = \frac{FP}{FP + TN}$$

A commonly used performance measure is the Area Under the Curve (AUC) of the ROC curve. The AUC is a numerical value that ranges from 0 to 1, where a value close to 1

TABLE 4. Configuration of Pre-trained Language Models used in the experimentation.

Method	Details of the model
FeedForwrad Neural Network	2 ReLU activated layers of size 256 and 32.
BERT (base)	110M parameters, Comprising 12 layers, with 768 hidden units and 12 attention heads, this model is trained on English text that has been converted to lowercase.
BERT (large)	340M parameters, Comprising 24 layers, with 1024 hidden units and 16 attention heads, this model is trained on English text that has been converted to lowercase.
RoBERTa	125M parameters,12-layer, 768-hidden, 12-heads. RoBERTa use the BERT-base architecture
DistilBERT	Featuring a 6-layer architecture with 768 hidden units and 12 attention heads, encapsulates a total of 66 million parameters. This model represents a distilled version derived from the 'bert-base-uncased' checkpoint of the BERT framework
XLM-RoBERTa	With 277.45 million parameters, the model features a configuration of 12 layers, 768 hidden states, and 3072 feed-forward hidden states, along with 8 attention heads. It is trained on 2.5 TB of freshly curated, clean CommonCrawl data spanning 100 languages
GPT-3	760M parameters, Comprising 24 layers, with 1536 hidden units and 16 attention heads.

indicates a nearly perfect model. The AUC provides a concise summary of the model's overall performance in distinguishing between positive and negative instances.

B. MODELS CONFIGURATION

We use the library *transformers* to implement the models from HuggingFace [1]. At the last stage of training, we identify the most effective model from the development set, assessing its performance after each epoch. This model is then applied to the test set for further evaluation and analysis. We have also provided the configuration of each PLM detailed in (see Table 4).

Following multiple heuristic tests, the hyper-parameters were carefully chosen to construct a final model with the highest performance in terms of the F1-score. The primary measure of system performance is the F1-score metric, which combines precision and recall metrics into a single value. Contrary to the traditional metric of accuracy, the F1-score offers a more comprehensive understanding of a model's proficiency in correctly classifying both Positive and Negative classes. We conducted the experiments on Google Colab Pro, utilizing a GPU accelerator (A100) and requiring high RAM capacity. However, due to the limited number of samples, each experiment had a runtime of approximately 20 minutes to 6 hours.

VI. RESULTS

To assess the solution's performance, we carried out multiple experiments using various forms of text representation and a Feedforward Neural Network on top. Furthermore, the FNN architecture was trained using supervised learning in a binary classification, with the Positive class representing Bots and the Negative class representing Human-created tweets. During this experimental evaluation, six Pre-trained Language Models (PLMs) based transformers were tested: BERT (base and large), RoBERTa, DistilBERT, XLM-RoBERTa and GPT-3-based embedding. The goal was to compare each embedding method's performance and decide which one was the most efficient. The same data distribution was utilized in all experiments: 80% for training and 20% for testing. To evaluate the PLMs, five evaluation metrics were used: Accuracy, AUC, Recall, Precision, and the F1-score. Each metric offers a unique perspective on different facets of the model's behavior, and relying on a single metric might not capture the full understanding, contributing to a poor evaluation. Table 5 presents the results of experiments conducted to train a deep neural network using diverse tweet-level embeddings from various Pretrained Language Models (PLMs) on two datasets: TweepFake and fox8-23.

The first batch of PLMs represents fine-tuned models, that show very accurate performance. The second batch of PLMs represents baseline approaches without fine-tuning. Therefore, we can conclude that the results of the fine-tuned models were statistically better than the baseline. The best result obtained for each metric is shown in bold. For the TweepFake dataset, BERT (large) and RoBERTa stand out with the highest fine-tuned accuracy (0.9407 and 0.9053), AUC (0.9234 and 0.9660), and F1-score (0.8887 and 0.9029). Similarly, for the fox8-23 dataset, GPT-3 achieves the highest fine-tuned accuracy (0.9307) and F1-score (0.93). The fine-tuned models, refined through iterative training on the bot datasets, exhibited a discernible enhancement in performance metrics. Notably, their capacity to discern intricate patterns and nuanced features surpassed that of the baseline models. This divergence was particularly pronounced in tasks demanding a nuanced understanding of contextual information, where the fine-tuned models demonstrated a marked superiority.

Conversely, the baseline models, while proficient, encountered challenges in extrapolating domain-specific intricacies. As we traverse the intricacies of this comparative evaluation, the fine-tuned models emerge as formidable contenders, showcasing the potential for tailored training to yield models adept at navigating the intricacies of our specific application domain. In addition, baseline models are not pre-entrained in social network jargon, so their performance is not good.

By analyzing these findings, it is evident that the embedding methods utilizing transformers such as RoBERTa, and particularly the fine-tuned GPT-3, significantly enhanced the F1-score. As we can see, the improvement rate in F1-score for RoBERTa on the TweepFake dataset is approximately 13.99%, demonstrating a significant enhancement in its ability to balance precision and recall after fine-tuning. Similarly, GPT-3 on the fox8-23 dataset shows an even higher improvement rate of 19.98% in F1-score.

[1]The largest community for sharing open-source pre-trained transformer models: https://huggingface.co/models

Once fine-tuned on the TweepFake dataset, the RoBERTa model demonstrates a notable enhancement in performance, achieving a classification AUC of approximately 96.60% for the most efficient PLM model, which also has a reduced parameter count. It is worth noting that different configurations of Pre-trained Language Models, specifically varying the dimensionality of the word embedding space, do not significantly impact performance. However, there is a general trend indicating that higher dimensionalities tend to result in slightly better performance. While the improvement may be subtle, increasing the dimensionality of the word embedding space appears to have a positive effect on overall performance.

Unlike BERT-based models, the GPT-3 language model has been partially pre-trained on social media data, making it more suitable for generating text in that domain.

Meanwhile, XLM-RoBERTa, a cross-lingual pre-trained model, attained a classification performance of 0.8821 (F1-score) on the *TweepFake* dataset. RoBERTa, which is pre-trained using English text, achieved an F1-score of 0.9029 on the same dataset, outperforming XLM-RoBERTa by approximately 2.36%. In the fox8-23 dataset, RoBERTa generally outperforms XLM-RoBERTa across most metrics, especially in accuracy, AUC, precision, and F1-score. Both models perform similarly in recall when fine-tuned, but RoBERTa's higher precision and F1-score suggest it is more reliable in correctly classifying both positive and negative instances. This comparison highlights the effectiveness of RoBERTa's pre-training on English text compared to the multilingual approach of XLM-RoBERTa.

To provide an overview of how the classification outcomes are distributed, we showcase the confusion matrix (see Table 6) of PLMs used as text embedding to classify tweets. A confusion matrix, typically utilized in assessing the efficacy of a classification model, is a tabular representation that compares the model's predictions against the actual values in a test dataset. It provides a visual representation of how well the algorithm performs. In the TweepFake dataset, RoBERTa and XLM-RoBERTa show strong performance, with RoBERTa achieving a high TP (2312) and low FP (229), while XLM-RoBERTa has the highest TP (2338) but also a high FN (400). For the fox8-23 dataset, GPT-3 and DistilBERT outperform other models, with GPT-3 showing the highest TP (5504) and TN (5665), and the lowest FP (426) and FN (405), indicating superior classification capability. Overall, the results indicate that fine-tuning significantly enhances model performance, with GPT-3 and DistilBERT emerging as top performers across both datasets.

To demonstrate the superiority of PLMs over traditional word embeddings, initially, we trained the LSTM classifier using the previous datasets, employing pre-trained embeddings and contrasting these with embeddings generated by PLMs. Furthermore, we utilized the identical preprocessing script for preparing the dataset. This preprocessing involves substituting URLs, numbers, user mentions, hashtags, and certain ASCII emoticons with their respective tags. The outcomes of this evaluation are displayed in Table 7. Overall, the findings suggest that fine-tuned pre-trained language models demonstrate superior performance compared to traditional pre-trained and contextualized embeddings when integrated with BiLSTM for bot detection. This supports the notion that language models effectively grasp the unique characteristics of social media and bot language, or at the very least, exhibit sufficient adaptability to generalize effectively during fine-tuning within this context. The architecture of the BiLSTM comprises an embedding layer, the BiLSTM layer with 128 units, and a fully connected layer that uses sigmoid as an activation function to predict the probability of each tweet being written by a bot or a human.

Fig. 6 illustrates DistilBERT's decision-making process in classifying a specific tweet as human-generated. This SHAP plot begins with a base value, representing the average model output across the test dataset used for comparison. The output value $f(x)$ is the actual model output for this specific instance. In this case, the output value is approximately 0.409, and it will be compared to the base value to ascertain the impact of each feature. The colors indicate the direction of the feature's effect. Typically, red will denote a positive contribution, and the other color (e.g., blue) will denote a negative contribution to the prediction. We observed that the word "the" contributed positively to the bot classification. In contrast, the inclusion of personal names like 'justin' and cognitive verbs like 'think' positively influenced the model's prediction of human origin. These findings suggest that DistilBERT, through its fine-tuning on *TweepFake* dataset, has learned to discern nuanced linguistic cues that can distinguish between human and bot-generated text.

Utilizing the XLM-RoBERTa architecture, renowned for its cross-lingual capabilities, we visualized the attention mechanism for the bot-generated tweet "nahhhhh all yee white people that". The attention plot from layer 2, head 2 (see Fig. 7), illustrates a pronounced focus on the [CLS] token, which aggregates the sequence representation. Remarkably, the token '##e' garners substantial attention, hinting at its potential significance within the tweet's context. The visualization also highlights the model's ability to process subword elements, such as '##hh' and '##e' demonstrating how XLM-RoBERTa attends to both complete and fragmented lexical units to infer meaning. This pattern of attention underscores the token's relevance in the model's interpretative process and could suggest linguistic features that are characteristic of bot-like communication, as learned during the model's training phase. Interestingly, a considerable amount of attention is directed towards special tokens such as [SEP] and [CLS], with the initial layers predominantly concentrating on the [CLS] token. Conversely, in the later layers, there is a tendency for the [SEP] token to receive more substantial attention. The insights gained here contribute to understanding how advanced transformer-based models, like XLM-RoBERTa, analyze and attribute significance to different text components, which is critical for enhancing bot detection algorithms.

LLM-based prompting provides human-readable explana-

TABLE 5. The outcomes of conducted experiments for training the deep neural network using diverse tweet-level embeddings derived from PLMs. These results are from the validation dataset. For each batch of models, we highlighted the best metrics

PLMs	Dataset	Accuracy	AUC	Recall	Precision	F1-score	Accuracy	AUC	Recall	Precision	F1-score
		Fine-tuned					**Baseline**				
BERT (base)	TweepFake	0.8989	0.9365	0.8709	**0.9210**	0.8907	0.7890	0.8813	0.7655	0.7878	0.7799
BERT (large)		**0.9407**	0.9243	**0.9240**	0.8623	0.8887	**0.8165**	**0.9145**	0.8322	0.8045	**0.8121**
RoBERTa		0.9053	**0.9660**	0.9176	0.8942	**0.9029**	0.7839	0.8920	**0.8687**	0.7405	0.7921
XLM-RoBERTa		0.8823	0.9642	0.8823	0.8846	0.8821	0.7934	0.8943	0.7934	0.7963	0.7930
DistilBERT		0.875	0.8743	0.8749	0.8582	0.8749	0.6316	0.6582	0.5598	0.6491	0.5918
GPT-3		0.8762	0.9307	0.8703	0.8918	0.8809	0.7571	0.7859	0.7980	**0.7545**	0.7756
BERT (base)	fox8-23	0.9150	0.9703	0.8880	**0.9387**	0.9075	0.7895	0.8771	0.7982	0.7845	0.7672
BERT (large)		0.9264	0.9083	0.9147	0.8732	0.8614	0.8125	0.8935	0.8391	**0.7967**	0.7941
RoBERTa		0.8566	0.8873	0.8696	0.8439	0.8566	0.7744	0.8682	0.7942	0.7588	0.7761
XLM-RoBERTa		0.8087	0.8683	0.8429	0.7846	0.8127	0.7714	0.8568	0.8468	0.7313	0.7848
DistilBERT		0.9284	**0.9768**	0.9295	0.9253	0.9274	**0.8234**	**0.9034**	**0.8617**	0.7963	**0.8277**
GPT-3		**0.9307**	0.9740	**0.9314**	0.9281	**0.9298**	0.7596	0.8360	0.8405	0.7189	0.7750

TABLE 6. Confusion Matrix Results for Different Models Across Datasets

Model	TweepFake				fox8-23			
	TP	FP	TN	FN	TP	FP	TN	FN
BERT (base)	2211	330	2324	175	5573	714	5286	427
BERT (large)	2259	368	2051	362	5035	1284	4715	966
RoBERTa	2312	229	2249	250	5139	950	770	5141
XLM-RoBERTa	2338	203	2099	400	4981	1367	4724	928
DistilBERT	2211	330	2224	275	5494	463	5628	415
GPT-3	2310	279	2108	343	5504	426	5665	405
glove.twitter.200d + BiLSTM	2224	490	1965	361	4428	452	6669	451
GoogleNews.300d + BiLSTM	2210	438	2018	374	4459	472	6648	421

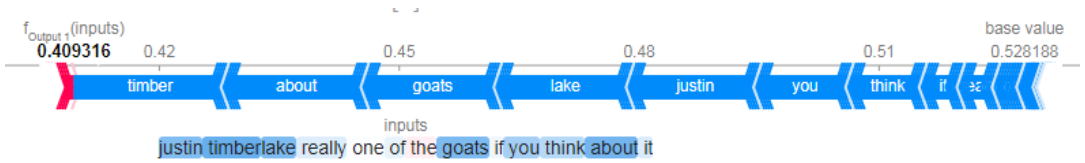


FIGURE 6. Interpreting classification decision of DistilBERT model fine-tuned with Dataset TweepFake for human-generated tweet

TABLE 7. Evaluation of BiLSTM Classifier Using Pre-Trained Embeddings

Embeddings	Dataset	Accuracy	F1-score
glove.twitter.200d	TweepFake	0.8591	0.7390
	fox8-23	0.9066	0.8365
GoogleNews.300d	TweepFake	0.8942	0.7844
	fox8-23	0.8816	0.8035

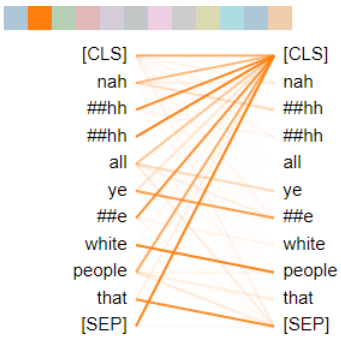
TABLE 8. Zero-Shot Example for Bot Detection Explanation with Feedback Justification applied by GPT-3

Prompt: The following social media tweet has been classified as bot. Explain this classification.
Tweet: yo this fall weather is poison and socially progressiveâ€ that sack, sis twitter activists, and i just me?
Feedback: The tweet exhibits several characteristics that are common in bot-generated content, including unusual phrasing and syntax errors ("socially progressiveâ€", "that sack, sis"). These irregularities suggest automated content generation rather than human speech. Additionally, the tweet seems to randomly combine different topics, which is a common trait in bot-generated content designed to engage different audiences without coherent context.

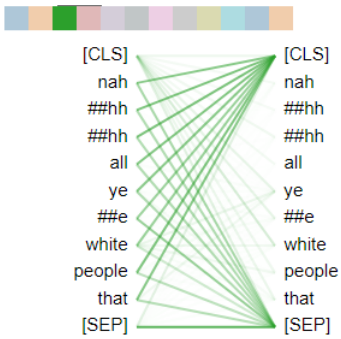
tions by interpreting the model's decision-making process through tailored prompts. By generating natural language explanations for each classification, we gained insights into the model's decision-making processes. The LLM-based explanations highlighted specific linguistic patterns and anomalies that contributed to the classification of content as bot-generated or human-generated. Given a social media post classified as bot-generated, automatically generate textual feedback that explains the classification to the user. Building on the strong abilities of LLMs in many text-generation tasks, this work examines how well LLMs can generate explanations for bot detection by using zero-shot prompting strategy [95]. This involves asking the LLM to perform the task without providing any examples. The model relies solely on its pre-existing knowledge. Table 8 demonstrates the effectiveness of GPT-4 in providing a zero-shot explanation for why a given tweet was classified as bot-generated.

VII. DISCUSSION

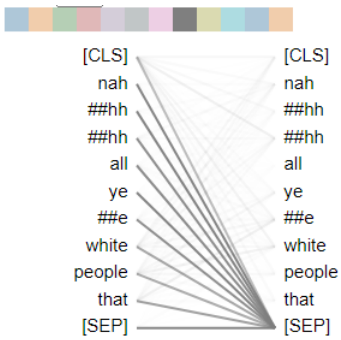
The findings of this study indicate that various detectors, including those based on deep learning methods and transfer learning using transformer architectures, face greater challenges in accurately detecting deepFake tweets compared to human-written ones. The capability of newer generative



(a) Layer 2, Head 2



(b) Layer 4, Head 3



(c) Layer 12, Head 8

FIGURE 7. XLM-RoBERTa's Attention Distribution Across Different Layers for bot-generated tweet

models, especially those based on transformer architectures like GPT. These advanced models demonstrate an enhanced ability to create text that mirrors human writing, outperforming traditional methods such as Recurrent Neural Networks (RNNs). A manual examination of numerous tweets from both GPT-2 and RNN sources revealed that tweets from the former were more challenging to categorize as bot-generated. However, further research is needed to thoroughly investigate the human-like nature of tweets generated by various generative methods by soliciting feedback from individuals. In the comparative analysis, we utilized the same dataset created by Fagni ***et al.*** [53] to evaluate the efficacy of the bot detection method. This direct comparison on a uniform dataset provided a clear and unbiased assessment of performance. The proposed method demonstrated superior accuracy and efficiency in identifying bots, highlighting the advancements in algorithmic design and data processing techniques. This consistent dataset usage allowed for more accurate benchmarking, underscoring the enhanced capabilities of the proposed approach in bot detection. The method achieved an F1-score of 90.29%, which outperforms the score of 89.6% obtained by Fagni ***et al.***. On the other hand, for the fox8-23 dataset, we achieved an F1-score of 93%, surpassing the 84% score reported by Yang and Menczer [96]. The difficulty with PLMs arises from their training on well-structured and grammatically correct text sources such as Wikipedia, news articles, or books. In contrast, social media posts are typically brief and replete with acronyms, hashtags, user mentions, URLs, and spelling errors. The goal is to detect bots within Twitter, as the text produced by these automated agents may differ significantly from the text in the sources used for training the models. The evaluation results presented in Table 7 indicate that the BiLSTM classifier using GloVe embeddings excels on the fox8-23 dataset, achieving 90.66% accuracy and an 83.65% F1-score. Conversely, for the TweepFake dataset, GoogleNews embeddings perform better, with 89.42% accuracy and a 78.44% F1-score, highlighting the importance of dataset-specific embedding selection. Unlike Word2Vec and GloVe, which generate a single embedding vector for each word based on the entire corpus (resulting in a static representation), transformer-based PLMs provide dynamic embeddings. This means that the representation of a word changes based on its context within a sentence, allowing the model to capture meanings specific to the sentence's context. For example, the word "bank" would have different embeddings when used in "river bank" vs. "bank account".

Both LLM-based prompting and SHAP have their unique strengths and weaknesses in the context of bot detection. LLM-based prompting excels in providing context-aware, human-readable explanations that enhance the interpretability and accessibility of model decisions. However, it requires careful prompt design and can sometimes produce inconsistent results. On the other hand, SHAP provides precise, mathematically grounded explanations of feature importance, ensuring robustness and consistency across different models and datasets. It is widely accepted for its transparency and quantifiable insights but may require domain expertise to interpret effectively.

This study has significant social implications by enhancing the accuracy and reliability of bot detection, thereby bolstering the trustworthiness of online environments and mitigating the spread of misinformation. Theoretically, this research advances the understanding of transformer-based models fine-tuned for specific tasks like social bot detection, demonstrating their effectiveness in handling complex linguistic patterns and distinguishing between human and bot-generated content. These findings contribute to the broader fields of natural language processing and machine learning, showcasing the potential of transformer-based models in real-world applications.

VIII. CONCLUSION AND FUTURE WORK

This study offered a secure approach for Twitter bot identification. The model we developed was built using pre-trained language models (PLMs). The proposed model's first goal was to use cutting-edge NLP Transformers to generate dense and significant representations of tweets. The subsequent objective was to employ a Deep Neural Network approach to construct a robust classifier capable of accurately identifying bot tweets in English with a high level of precision. For this task, various Pre-trained Language Models were used (BERT, GPT-3, and Roberta). The fine-tuned RoBERTa model achieved state-of-the-art outcomes on the TweepFake dataset, achieving an F1-score of 90.29%. Additionally, the fine-tuned GPT-3 model achieved an impressive F1-score of 93% on the fox8-23 dataset. While traditional embedding models like Word2Vec and GloVe have played pivotal roles in advancing natural language processing, they exhibit certain limitations when applied to complex tasks such as detecting bots in social media. These models generate static embeddings, which fail to capture the contextual nuances of language as effectively as newer Pretrained Language Models (PLMs) like BERT. Employing explainability techniques such as SHAP in bot detection has proven essential for understanding and trusting transformer-based models. These methods illuminate how models differentiate bots from humans by quantifying the contribution of each feature to the model's predictions. Furthermore, integrating LLM-based prompting provides detailed, context-aware explanations in natural language, making the decision-making process more accessible and transparent. Future work will focus on refining the embedding mechanism to support languages other than English. This will enable the model to effectively handle text data in a wider range of languages and expand its applicability to diverse linguistic contexts. Furthermore, we plan to integrate datasets TwiBot-20 [68] and TwiBot-22 [66], enabling the combination of content-based and graph-based classification methods. This integration aims to provide a more robust approach to detecting bot activity.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

REFERENCES

[1] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 3, pp. 1–41, 2020.

[2] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection," in *Advanced Information Networking and Applications: Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, pp. 1341–1354, Springer, 2020.

[3] M. Aljabri, R. Zagrouba, A. Shaahid, F. Alnasser, A. Saleh, and D. M. Alomari, *Machine learning â€˘ based social media bot detection : a comprehensive literature review*. Springer Vienna, 2023.

[4] A. Ali and A. M. Syed, "Cyberbullying detection using machine learning," *Pakistan Journal of Engineering and Technology*, vol. 3, no. 2, pp. 45–50, 2020.

[5] Jason Wise, "TWITTER BOTS PERCENTAGE: HOW MANY BOTS ARE ON TWITTER?," *Earth Web*.

[6] W. Yue and L. Li, "Sentiment analysis using word2vec-cnn-bilstm classification," in *2020 seventh international conference on social networks analysis, management and security (SNAMS)*, pp. 1–5, IEEE, 2020.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[8] S. S. Roy, A. I. Awad, L. A. Amare, M. T. Erkihun, and M. Anas, "Multimodel phishing url detection using lstm, bidirectional lstm, and gru models," *Future Internet*, vol. 14, no. 11, p. 340, 2022.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[10] A. Moreo, A. Esuli, and F. Sebastiani, *Word-class embeddings for multiclass text classification*, vol. 35. Springer US, 2021.

[11] M. Tezgider, B. Yildiz, and G. Aydin, "Text classification using improved bidirectional transformer," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 9, p. e6486, 2022.

[12] S. Kumar and A. Solanki, "An abstractive text summarization technique using transformer model with self-attention mechanism," *Neural Computing and Applications*, pp. 1–20, 2023.

[13] H. Chouikhi and M. Alsuhaibani, "Deep transformer language models for arabic text summarization: A comparison study," *Applied Sciences*, vol. 12, no. 23, p. 11944, 2022.

[14] Y. Kawara, C. Chu, and Y. Arase, "Preordering encoding on transformer for translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 644–655, 2020.

[15] M. Rikters and M. Pinnis, "Debugging translations of transformer-based neural machine translation systems," *Baltic Journal of Modern Computing*, vol. 6, no. 4, pp. 403–417, 2018.

[16] Z. Abbasiantaeb and S. Momtazi, "Entity-aware answer sentence selection for question answering with transformer-based language models," *Journal of Intelligent Information Systems*, vol. 59, no. 3, pp. 755–777, 2022.

[17] Y. Li, R. M. Wehbe, F. S. Ahmad, H. Wang, and Y. Luo, "A comparative study of pretrained language models for long clinical text," *Journal of the American Medical Informatics Association*, vol. 30, no. 2, pp. 340–347, 2023.

[18] Y. Pengbo, P. Weimin, Z. Haijun, and C. Degang, "Identifying clickbait with bert-biga model," *Data Analysis and Knowledge Discovery*, vol. 5, no. 6, pp. 126–134, 2021.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[20] T. Han, Z. Zhang, M. Ren, C. Dong, X. Jiang, and Q. Zhuang, "Text emotion recognition based on xlnet-bigru-att," *Electronics*, vol. 12, no. 12, p. 2704, 2023.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[22] A. Hern and J. Bhuiyan, "Openai says new model gpt-4 is more creative and less likely to invent facts," *The Guardian*, 2023.

[23] P. E. Love, W. Fang, J. Matthews, S. Porter, H. Luo, and L. Ding, "Explainable artificial intelligence (xai): Precepts, models, and opportunities for research in construction," *Advanced Engineering Informatics*, vol. 57, p. 102024, 2023.

[24] L. Luceri, T. Braun, and S. Giordano, "Analyzing and inferring human real-life behavior through online social networks with social influence deep learning," *Applied network science*, vol. 4.

[25] T. Dourado, "Who posts fake news? authentic and inauthentic spreaders of fabricated news on facebook and twitter," *Journalism Practice*, vol. 17, no. 10, pp. 2103–2122, 2023.

[26] W. Shahid, Y. Li, D. Staples, G. Amin, S. Hakak, and A. Ghorbani, "Are you a cyborg, bot or human?—a survey on detecting fake news spreaders," *IEEE Access*, vol. 10, pp. 27069–27083, 2022.

[27] S. Li, Y. Cao, S. Liu, Y. Lai, Y. Zhu, and N. Ahmad, "Hda-ids: A hybrid dos attacks intrusion detection system for iot by using semi-supervised cl-gan," *Expert Systems with Applications*, vol. 238, p. 122198, 2024.

[28] R. Abu Khurma, I. Almomani, and I. Aljarah, "Iot botnet detection using salp swarm and ant lion hybrid optimization model," *Symmetry*, vol. 13, no. 8, 2021.

[29] J. Sutton, "Health communication trolls and bots versus public health agencies' trusted voices," 2018.

[30] M. Zhang, Z. Chen, X. Qi, and J. Liu, "Could social bots' sentiment engagement shape humans' sentiment on covid-19 vaccine discussion on twitter?," *Sustainability (Switzerland)*, vol. 14, no. 9, 2022. Cited by: 3; All Open Access, Gold Open Access.

[31] Z. Ellaky, F. Benabbou, and S. Ouahabi, "Systematic literature review of social media bots detection systems," *Journal of King Saud University-Computer and Information Sciences*, 2023.

[32] U. Krzeszewska, A. Poniszewska-Marańda, and J. Ochelska-Mierzejewska, "Systematic comparison of vectorization methods in classification context," *Applied Sciences (Switzerland)*, vol. 12, no. 10, 2022. Cited by: 7; All Open Access, Gold Open Access.

[33] M. B. Torusdağ, M. Kutlu, and A. A. Selçuk, "Are we secure from bots? investigating vulnerabilities of botometer," in *2020 5th International Conference on Computer Science and Engineering (UBMK)*, pp. 343–348, IEEE, 2020.

[34] X. Du, S. Chen, Z. Liu, and J. Wang, "Multiple userids identification with deep learning," *Expert Systems with Applications*, vol. 207, p. 117924, 2022.

[35] G. K. Venkatesh, V. Srihari, R. Veeramani, R. Karthikeyan, and R. Anitha, "Http botnet detection using hidden semi-markov model with snmp mib variables," *International Journal of Electronic Security and Digital Forensics*, vol. 5, no. 3-4, p. 188 – 200, 2013. Cited by: 3.

[36] H.-S. Won, M.-J. Kim, D. Kim, H.-S. Kim, and K.-M. Kim, "University student dropout prediction using pretrained language models," *Applied Sciences*, vol. 13, no. 12, p. 7073, 2023.

[37] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, 2018.

[38] J. Tourille, B. Sow, and A. Popescu, *Automatic Detection of Bot-generated Tweets*, vol. 1. Association for Computing Machinery, 2022.

[39] H. Stiff and F. Johansson, "Detecting computer-generated disinformation," *International Journal of Data Science and Analytics*, vol. 13, no. 4, pp. 363–383, 2022.

[40] J. Pu, Z. Sarwar, S. M. Abdullah, A. Rehman, and Y. Kim, "Deepfake Text Detection : Limitations and Opportunities,"

[41] D. Martín-gutiérrez and A. B. Hernández, "A Deep Learning Approach for Robust Detection of Bots in Twitter Using Transformers," vol. 9, 2021.

[42] D. Dukic, D. Keca, and D. Stipic, "Are you human? detecting bots on twitter using BERT," *Proceedings - 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA 2020*, pp. 631–636, 2020.

[43] S. Kumar, S. Garg, Y. Vats, and A. S. Parihar, "Content Based Bot Detection using Bot Language Model and BERT Embeddings," *2021 5th International Conference on Computer, Communication, and Signal Processing, ICCCSP 2021*, pp. 285–289, 2021.

[44] Q. Guo, H. Xie, Y. Li, W. Ma, and C. Zhang, "Social bots detection via fusing bert and graph convolutional networks," *Symmetry*, vol. 14, no. 1, pp. 1–14, 2022.

[45] X. Liu, Y. Zhan, H. Jin, Y. Wang, and Y. Zhang, "Research on the Classification Methods of Social Bots," *Electronics (Switzerland)*, vol. 12, no. 14, 2023.

[46] M. Heidari and J. H. Jones, "Using BERT to Extract Topic-Independent Sentiment Features for Social Media Bot Detection," *2020 11th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2020*, pp. 0542–0547, 2020.

[47] Y. Wu, Y. Fang, S. Shang, J. Jin, L. Wei, and H. Wang, "A novel framework for detecting social bots with deep neural networks and active learning," *Knowledge-Based Systems*, vol. 211, p. 106525, 2021.

[48] L. Ilias and I. Roussaki, "Detecting malicious activity in Twitter using deep learning techniques," *Applied Soft Computing*, vol. 107, p. 107360, 2021.

[49] F. Zeng, Y. Sun, and Y. Li, "MRLBot : Multi-Dimensional Representation Learning for Social Media Bot Detection," 2023.

[50] S. Feng, Z. Tan, R. Li, and M. Luo, "Heterogeneity-Aware Twitter Bot Detection with Relational Graph Transformers," *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022*, vol. 36, pp. 3977–3985, 2022.

[51] A. Garcia-Silva, C. Berrio, and J. M. Gómez-Pérez, "An empirical study on pre-trained embeddings and language models for bot detection," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 148–155, 2019.

[52] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," in *Proceedings of the 26th international conference on world wide web companion*, pp. 963–972, 2017.

[53] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "TweepFake: About detecting deepfake tweets," *PLoS ONE*, vol. 16, no. 5 May, pp. 1–16, 2021.

[54] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," *Advances in neural information processing systems*, vol. 32, 2019.

[55] K.-C. Yang, O. Varol, C. A. Davis, E. Ferrara, A. Flammini, and F. Menczer, "Arming the public with artificial intelligence to counter social bots," *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 48–61, 2019.

[56] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1096–1103, 2020.

[57] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization," in *Proceedings of the international AAAI conference on web and social media*, vol. 11, pp. 280–289, 2017.

[58] F. Johansson, "Supervised classification of twitter accounts based on textual content of tweets. notebook for pan at clef 2019," in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop*, 2019.

[59] M. Mazza, S. Cresci, M. Avvenuti, W. Quattrociocchi, and M. Tesconi, "Rtbust: Exploiting temporal patterns for botnet detection on twitter," in *Proceedings of the 10th ACM conference on web science*, pp. 183–192, 2019.

[60] Z. Gilani, R. Farahbakhsh, G. Tyson, L. Wang, and J. Crowcroft, "Of bots and humans (on twitter)," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pp. 349–354, 2017.

[61] S. Cresci, F. Lillo, D. Regoli, S. Tardelli, and M. Tesconi, "Fake: Evidence of spam and bot activity in stock microblogs on twitter," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, 2018.

[62] S. Cresci, F. Lillo, D. Regoli, S. Tardelli, and M. Tesconi, "Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on twitter," *ACM Transactions on the Web (TWEB)*, vol. 13, no. 2, pp. 1–27, 2019.

[63] K. Lee, B. Eoff, and J. Caverlee, "Seven months with the devils: A long-term study of content polluters on twitter," in *Proceedings of the international AAAI conference on web and social media*, vol. 5, pp. 185–192, 2011.

[64] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015.

[65] S. Fakhraei, J. Foulds, M. Shashanka, and L. Getoor, "Collective spammer detection in evolving multi-relational social networks," in *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 1769–1778, 2015.

[66] S. Feng, Z. Tan, H. Wan, N. Wang, Z. Chen, B. Zhang, Q. Zheng, W. Zhang, Z. Lei, S. Yang, *et al.*, "Twibot-22: Towards graph-based twitter bot detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35254–35269, 2022.

[67] B. Gu, Z. Zhai, X. Li, and H. Huang, "Towards fairer classifier via true fairness score path," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3113–3121, 2022.

[68] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Twibot-20: A comprehensive twitter bot detection benchmark," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4485–4494, 2021.

[69] Z. Gilani, E. Kochmar, and J. Crowcroft, "Classification of twitter accounts into automated agents and human users," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, (New York, NY, USA), p. 489–496, Association for Computing Machinery, 2017.

[70] J. Wu, X. Ye, and Y. Man, "Bottrinet: A unified and efficient embedding for social bots detection via metric learning," in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, IEEE, 2023.

[71] F. Casillo, V. Deufemia, and C. Gravino, "Detecting privacy requirements from user stories with nlp transfer learning models," *Information and Software Technology*, vol. 146, p. 106853, 2022.

[72] J. Campos, A. Yee, and I. F. Vega, "Simplifying vgg-16 for plant species identification," *IEEE LATIN AMERICA TRANSACTIONS*, vol. 20, pp. 2330–2338, NOV 2022.

[73] P. Wang, F. Luo, L. Wang, C. Li, Q. Niu, and H. Li, "S-resnet: An improved resnet neural model capable of the identification of small insects," *FRONTIERS IN PLANT SCIENCE*, vol. 13, DEC 22 2022.

[74] Y. Liu, W. Che, Y. Wang, B. Zheng, B. Qin, and T. Liu, "Deep contextualized word embeddings for universal dependency parsing," *ACM TRANSACTIONS ON ASIAN AND LOW-RESOURCE LANGUAGE INFORMATION PROCESSING*, vol. 19, JAN 2020.

[75] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[76] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[77] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[78] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.

[79] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[80] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," *arXiv preprint arXiv:1704.04683*, 2017.

[81] F. S. Alrayes, M. Zakariah, M. Driss, and W. Boulila, "Deep neural decision forest (dndf): A novel approach for enhancing intrusion detection systems in network traffic analysis," *Sensors*, vol. 23, no. 20, 2023.

[82] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[83] V. Belle and I. Papantonis, "Principles and Practice of Explainable Machine Learning," 2021.

[84] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[85] E. Martínez García, A. Nogales Moyano, J. Morales Escudero, Á. J. García Tejedor, *et al.*, "A light method for data generation: a combination of markov chains and word embeddings.," 2020.

[86] A. R. Tayal and M. A. Tayal, "Darnn: Discourse analysis for natural languages using rnn and lstm.," *International Journal of Next-Generation Computing*, vol. 12, no. 5, 2021.

[87] W. Fang, T. Jiang, K. Jiang, F. Zhang, Y. Ding, and J. Sheng, "A method of automatic text summarisation based on long short-term memory," *International Journal of Computational Science and Engineering*, vol. 22, no. 1, pp. 39–49, 2020.

[88] D. Demırcı, C. Acarturk, *et al.*, "Static malware detection using stacked bilstm and gpt-2," *IEEE Access*, vol. 10, pp. 58488–58502, 2022.

[89] L. Zhang and Q. Yan, "Detect malicious websites by building a neural network to capture global and local features of websites," *Computers and Security*, vol. 137, 2024. Cited by: 0.

[90] C. S. Devi and B. S. Purkayastha, "An empirical analysis on statistical and neural machine translation system for english to mizo language," *International Journal of Information Technology (Singapore)*, vol. 15, no. 8, p. 4021 – 4028, 2023. Cited by: 0.

[91] K. Awadh and A. AKBAŞ, "Intrusion detection model based on tf. idf and c4. 5 algorithms," *Politeknik Dergisi*, vol. 24, no. 4, pp. 1691–1698, 2021.

[92] H. G. M. L. A. A. M. K. N. A. M. A. S. A. A. M. Anwer Mustafa Hilal, Aisha Hassan Abdalla Hashim, "Spotted hyena optimizer with deep learning driven cybersecurity for social networks," *Computer Systems Science and Engineering*, vol. 45, no. 2, pp. 2033–2047, 2023.

[93] J.-W. Baek and K.-Y. Chung, "Multimedia recommendation using word2vec-based social relationship mining," *Multimedia Tools and Applications*, vol. 80, pp. 34499–34515, 2021.

[94] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[95] N. Knoth, A. Tolzin, A. Janson, and J. M. Leimeister, "Ai literacy and its implications for prompt engineering strategies," *Computers and Education: Artificial Intelligence*, vol. 6, 2024. Cited by: 0; All Open Access, Gold Open Access.

[96] K.-C. Yang and F. Menczer, "Anatomy of an ai-powered malicious social botnet," *ArXiv*, vol. abs/2307.16336, 2023.

AMINE SALLAH He is currently pursuing his PhD at the MAIS Laboratory, Faculty of Sciences and Techniques, Moulay Ismail University of Meknes, Morocco. He earned his MS degree in business intelligence and image processing from the Faculty of Sciences and Techniques of Errachidia in 2020. Presently, he serves as a high school computer science teacher at the Ministry of Education in Morocco. His research is focused on artificial intelligence and its applications in cybersecurity.

EL ARBI ABDELLAOUI ALAOUI received the PhD degree in Computer Science in 2017 from Faculty of Sciences and Technology - Errachidia, University of Moulay Ismaïl, Meknès, Morocco. Prior to this, he received the Master's degree in Telecommunication in 2013 from the National School of Applied Sciences, University of Sidi Mohamed Ben Abdallah, Fès, Morocco. He is currently a Professor at Moulay Ismail University (UMI), Meknes, Morocco. His research publications include mainly IA and Machine learning, wireless networking, Ad hoc networking, DTN networks, game theory, Internet of Things (IoT) and Smart cites.

SAID AGOUJIL received the PhD and MS degrees in mathematics from Faculty of Sciences and Technology of Marrakech (FSTM), Morocco in 2008 and 2004, respectively. He is currently a full professor at the Department of Computer Science, École Nationale de Commerce et de Gestion, My Ismail University, El Hajeb, Morocco. His current research interests include numerical analysis, wireless networks, linear algebra, and speech coding.

MUDASIR AHMAD WANI received the Master of Computer Applications (M.C.A.) and M.Phil. degrees in data mining from the University of Kashmir (UoK), in 2012 and 2014, respectively, and the Ph.D. degree in computer science from Jamia Millia Islamia (A Central University), New Delhi, India, in 2019. He pursued his postdoctoral research with the Norwegian Biometrics Laboratory, Norwegian University of Science and Technology (NTNU), Norway. He was a Lecturer and a Researcher with the Department of Information Security and Communication Technology (IIK), NTNU. He is currently a Researcher in NLP with Prince Sultan University, Saudi Arabia. He is actively involved in organizing and reviewing international conferences, workshops, and journals. His research interests include the extraction and analysis of social data and the application of different statistical and machine/deep learning techniques in developing prediction models. He was a recipient of the Alain Bensoussan Fellowship Award under the European Research Consortium for Informatics and Mathematics (ERCIM), Sophia Antipolis Cedex, France

MOHAMED HAMMAD received his Ph.D. degree in 2019, the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. He is an assistant professor in the Faculty of Computers and Information, Menoufia University, Egypt. He is currently a researcher in EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University. His research interests include Biomedical Imaging, Bioinformatics, Cyber Security, IoT, Computer Vision, Machine Learning, Deep Learning, Pattern Recognition, and Biometrics. He has published more than 50 papers in international SCI-IF journals. Furthermore, he has served as an Editor Board member in PLOS ONE journal, an Editor Board member in BMC Bioinformatics journal, an Associate Editor in IJISP, a Topics Board editor in Forensic Sciences (MPDI) journal, a guest editor in many international journals such as IJDCF, Sensors (MDPI) and Information (MDPI). Reviewer of more than 500 papers for many prestigious journals and listed in the top 2

AHMED A. ABD EL-LATIF (Senior Member, IEEE) graduated with a distinguished Ph.D. from Harbin Institute of Technology, China, in 2013. Since 2013 carried out a number of successful research projects and grants in Egypt, Russian Federation, Saudi Arabia, China, Malaysia, and Tunisia. He is a staff member at Menoufia University, Egypt, and at Prince Sultan University, Saudi Arabia. In more than 18 years of his professional experience, he published about 300 papers in journals/conferences including 12 books with over 10000 citations. Since 2022 is a head of MEGANET 6G lab research in Russian Federation. Abd El-Latif is a vice chair of EIAS research lab, founder and deputy director of the center of excellence in Quantum & Intelligent Computing (Prince Sultan University, Saudi Arabia). Abd ElLatif was honored with several awards, including State Encouragement Award in Engineering Sciences 2016, Arab Republic of Egypt; the best Ph.D. student award from Harbin Institute of Technology, China 2013; Young scientific award, Menoufia University, Egypt 2014. Abd El-Latif is serving as chair/co-chair of many Scopus/ EI conferences. He is the EIC of International Journal of Information Security and Privacy. Series editor of Quantum Information Processing and Computing, and series editor of Advances in Cybersecurity Management. Also, academic editor/ associate editor for many indexed journals (WoS and Scopus journals' quartile ranking). His research interests include Quantum communications and cryptography, Cybersecurity, Artificial Intelligence of Things, AI-Based image processing, Information hiding, and Applications of dynamical systems (discrete-time models: chaotic systems and quantum walks) in cybersecurity

YASSINE MALEH (Senior Member, IEEE) received the dual Ph.D. degree in computer sciences management. He is currently an Associate Professor of cybersecurity and IT governance with Sultan Moulay Slimane University, Morocco. He is the Founding Chair of the IEEE Consultant Network Morocco and the Founding President of the African Research Center of Information Technology Cybersecurity. He has published over 100 papers (book chapters, international journals, and conferences/workshops), 23 edited books, and five authored books. He is a member of the International Association of Engineers IAENG and the Machine Intelligence Research Labs. He received the Publons Top 1Reviewer Award for the years 2018 and 2019. He was the Publicity Chair of BCCA 2019 and the General Chair of the MLBDACP 2019 Symposium and ICI2C 2021 Conference. He is the Editor-in-Chief of the International Journal of Information Security and Privacy and the International Journal of Smart Security Technologies (IJSST). He serves as an Associate Editor for IEEE ACCESS (2019 Impact Factor 4.098), the International Journal of Digital Crime and Forensics (IJDCF), and the International Journal of Information Security and Privacy (IJISP). He is a Series Editor of Advances in Cybersecurity Management (CRC Taylor Francis). He has served and continues to serve on executive and technical program committees and as a Reviewer for numerous international conferences and journals, such as Ad Hoc Networks (Elsevier), IEEE Network magazine, IEEE SENSORS JOURNAL, ICT Express, and Cluster Computing (Springer).