https://doi.org/10.31449/inf.v46i6.4081 Informatica **46** (2022) 143–158 **143**

Hybrid-MELAu: A Hybrid Mixing Engineered Linguistic Features Framework Based on Autoencoder for Social Bot Detection

Zineb Ferhat Hamida[1], Allaoua Refoufi[1], Ahlem Drif[1] and Silvia Giordano[2]
E-mail: zineb.ferhat@yahoo.com, allaoua.refoufi@univ.setif.dz, adrif@univ.setif.dz, silvia.giordano@supsi.ch
[1]Networks and Distributed Systems Laboratory, Department of Computer Science, University of Sétif 1, Sétif, Algeria
[2]Networking Lab, SUPSI University of Applied Sciences of Southern Switzerland Lugano, Switzerland

Keywords: social bots detection, natural language processing, autoencoder, recurrent neural network, feature engineering, classification

Received: March 17, 2022

Social bots are defined as computer algorithms that generate massive amounts of obnoxious or meaningful information. Most bot detection methods leverage multitudinous characteristics, from network features, temporal dynamics features, activities features, and sentiment features. However, there has been fairly lower work exploring lexicon measurement and linguistic indicators to detect bots. The main purpose of this research is to recognize the social bots through their writing style. Thus, we carried out an exploratory study on the effectiveness of only a set of linguistic features (17 features) exploitable for bot detection, without the need to resort to other types of features. And we develop a novel framework in a hybrid fashion of Mixing Engineered Linguistic features based on Autoencoders (Hybrid-MELAu). The semi-supervised Hybrid-MELAu framework is composed of two essential constituents: the features learner and the predictors. We establish the features learner innovated on two powerful structures: a) the first is a Deep dense Autoencoder fed by the Lexical and the Syntactic content (DALS) that represents the high order lexical and syntactic features in latent space, b) the second one is a Glove-BiLSTM autoencoder, which sculpts the semantic features; subsequently, we generate elite elements from the pre-trained encoder part from each latent space with transfer learning. We consider a sample of 1 Million from Cresci datasets to conduct our linguistic analysis comparison between the writing style of humans and bots. With this dataset, we observe that the bot's textual lexical diversity median is greater than the human one and the syntactic analysis based on speech-tagging shows a creative behavior in human writing style. Finally, we test the model's robustness on several public dataset (celebrity, pronbots-2019, and political bots). The proposed framework achieves a good accuracy of 92.22%. Overall, the results shown in this paper, and the related discussion, argue that it is possible to discern the differences between humans' and bots' writing styles based on an efficient linguistic deep framework.

Povzetek: V prispevku je opisana metoda za detekcijo pogovornih asistentov (botov) na osnovi jezkovnega stila.

1 Introduction

As a result of the invention of social media, many users are performing various acts that can produce incorrect information that propagates easily through the internet for different purposes [1]. Some try to deceive the reader or sway his perspective on a topic. Others are created from scratch with a tempting caption to enhance website traffic and visits. Recently, there have been several works girding fake news features analysis [2, 3, 4, 5, 6, 7]. A veritably complex task consists to supervise and investigate the diffusion's information sources and the nature of profit users. This is owing to users' aversion to disclosing their genuine identities, which is regarded illegal, or even these users can be nonhuman (social bots and cyborg use). We focus in this work on social bots detection as it became an efficient mechanism for fake news propagation that can get a negative impact on individuals and society.

When working with social bots detection, one of great challenges is features engineering. Multitudinous exploration pinpoints spambots through multi-feature approaches. Kosmajac et al [8] extracted a fingerprint of user behavior from the users' features to realize automated users. They applied machine learning algorithms to discover the social bots. However, the textual content itself is probable to be a crucial characteristic for social bots detection. It's therefore important to reach a way of representing the text that can capture the information necessary for acknowledging if the account is either human or bot. Wei et al [9] concentrated on distinguish between twitter accounts of both human and spambot, by building a BiLSTM network to efficiently capture the content features across tweets. Different from these works, our work focuses on developing a novel framework based on linguistic features to recognize the social bots through their writing styles. In other words, linguistics features can add significant value

to the retrieval information process.

Hence, we design a linguistically oriented framework that combines the embedding-based strength with the advantage offered by Autoencoder (AEs) in dimensionality reduction. The proposed architecture is separated into two segments: the features learner and a deep neural networks classifier. The feature learner aims at performing the feature extraction task due to a deep autoencoder based on dense layers and a BiLSTM autoencoder. We enhance the feature extractor: (i) by feeding the lexical and syntactic features to the first autoencoder to represent the high order features in latent space; (ii) constituting the semantic and the context features using the BiLSTM autoencoder; (iii) the merging of the two previous trained encoder blocks would generate a compacted data to get rid of any tangential information, and just concentrate on the highly essential characteristics which would discover human writing style patterns accurately. We summarized our contribution as follows:

- Extracting various feature sets that indicate the writing style for both humans and bots, to be able to compare and evaluate the performance of the social bot detection model enhanced by the distinct writing patterns in estimating the human and bot classes. The different writing style features are lexical features relying on text richness and diversity, syntactic features based on Pos-tagging, and semantic features that are extracted with word embeddings techniques.

- We develop Hybrid-MELAu: novel semi-supervised framework in a hybrid fashion mixing engineered linguistic features based on autoencoder. Therefore, we trained two autoencoders. The first is a deep dense autoencoder fed by the lexical and the syntactic features (DALS). The second one is a Glove word embedding BiLSTM autoencoder (GloVe-BiLSTM autoencoder), which effectively captures the semantic or the contextual features across tweets. Then, we stapled the trained encoder building blocks to generate elite characteristics from both latent spaces. The idea behind this combination is to complement one another; they successfully model the lexical, syntactic, and semantic knowledge. In low-dimensional spaces, this representation will become very efficient.

- We benefit from the profound features attained from encoders for transfer learning to discern differences in the writing styles of both humans and bots. The initialization of the classifiers with transferred features has improved the performance when modeling the bot detection task.

- We chain the Hybrid-MELAu output with six Recurrent Neural Network classifiers: SimpleRNN, BiRNN, LSTM, BiLSTM, GRU (Gate Recurrent Units), and BiGRU.

- Experiments were carried out with a real-world data set. Trials show that the introduced framework significantly achieves an accurate social bot detection.

The paper rest is categorized following this order: section 2 explain the objectives of this research, the third section delves into related works. Section 4 is devoted to the features extraction study and the most prominent measures for natural language texts. Section 5 presents the proposed semi-supervised framework for building high-performance bot detection models. Section 6 details our architectures results applied on real-world datasets. Then, a beneficial discussion is provided in section 7. Eventually, the conclusion is provided in Section 8.

2 Research objectives

- Our focus was on investigating the bot writing style to confirm whether it is possible or not to obtain a competitive detection performance using just a set of relevant linguistic features, unlike the majority of work on bot detection that have investigated a bigger set of features without taking into account the specific type-token ratio and the vocabulary knowledge.

- Since a successful bot can use a linguistic approach based on the linguistic structure, we aim at digging deeply to show that the linguistics features can add significant value to differentiate human accounts from bot accounts. Our exploratory study address the following writing style features analysis: lexical features relying on text richness and diversity, syntactic features based on Pos-tagging, and word embeddings approaches are used to extract semantic features.

- Bot detection has been implemented using a variety of deep learning and machine learning methods [10, 11], but there is still much work to be done. In fact, exploiting only the bots automatic writing style behaviors is a challenging task because their combination produce incomplete, unstructured, and noisy data. Therefore, we will develop a hybrid deep learning approach based only on linguistic features that can improve the detection performance.

3 Related work

The bots are hefty threats on social media credibility. Understanding of bot accounts behaviors in social platforms is of pivotal importance to enable its detection. El-Mawass at al [12] created a Markov Random Field on a graph of similar users and utilize state-of-the-art classifiers to infer previous beliefs, they used Loopy Belief Propagation to get later predictions about the user. Yang et al [13] displayed that most common political messages on Twitter issued by a small group of hyperactivity accounts. Authors noted that overactive users are more probably to spread low-integrity

information for quotidian users, and they tend to show suspicious behaviors often associated with false accounts automation. Yang et al [14] published a web application named bot electioneering volume (BEV) that notifies the level of bot practices and presents the subjects addressed by them on twitter per diem. Many studies have focalized on collective and not genuine activities for harmful accounts to detect consistent campaigns and disseminate suspected content. Bot2Vec is an innovative approach to identifying bots/spammers offered by Pham et al [15]. The approach is based on the local neighborhood relationships and the community internal structure of human nodes. Nizzoli et al [16] demonstrated that invite link exchange is a proxy for homophily and habitual goals between the involved agents and characteristic patterns related to deceptive schemes. In the work of Giglietto et al [17], The researchers devised a mechanism for recognizing coordinated link-sharing behavior (CLSB). By scanning URLs published by public groups, pages, and validated profiles on Facebook, this approach generates and maintains lists of sources that could be troublesome. Luceri et al [18] studied bots and humans virtual attitude and compared their communicating activities. Based on the nature and the quarrel within the online discussion, Luceri et al [19] identified several accounts classes. The researchers concluded that in the political debate the hyperactive bots played a consequential role in the news diffusion.

Bots accounts are a problem on social media since they can inveigle information, disseminate misinformation,and inflate unverified news, which can alter the social media analyses results. Generally Social bot detection approaches are supervised. Ferrera et al [20] use an vast range of characteristics (Timing of tweets, network of tweet interactions, meaning, language, and emotions) and create a k-nearest neighbor with dynamic time warping (KNN-DTW) method for online bot detection. Cresci et al [21] work founded on DNA inspired fingerprinting coding to investigate social media user behavior in temporal dimension. In the work of Kudugunta et al [22], The authors pulled contextual information extracted from user metadata and provided as additional entry to LSTM deep neural network that analyse tweet text in order to identify bots. Yang et al [23] suggested a framework that utilizes minimum metadata account while focusing just on user profile information. In Heidari [24] work,the authors proposed a Bidirectional Encoder model for tweets sentiment classification to determine features from topic-independent for bot detection model. Sayyadiharikandeh et al [25] suggested a supervised learning approach that uses the maximum rule to combine the decisions of specialized classifiers. The suggested method is included in Botometer's latest version (v4), a commonly used tool for detecting social bots. To categorize tweets as bot tweets or not, a neural network ensemble of CNN and LSTM models with BERT embedding was developed by Kumar et al [26] which is based on the tweets' textual content. Gaurav et al [27] pinpointed account patterns types using machine learning mechanisms and provides intelligent clues that may be utilized as a robustness gauge for several systems. Several machine learning approaches for detecting malicious users have been suggested on Praveena [28] work based on glow worm optimization technique to in order to deal with a small set of features. The authors employed generalized regression neural network to train these features. Also, Chakraborty et al [29] proposed an innovative method for categorizing Twitter user accounts as valid or illegitimate, by using a combination of features engineered from user Metadata.The authors integrated graph centrality values and graph embedding generating from the followers-followings graph.

In addition to these works based on supervised and unsupervised approaches, present studies utilize a semi-supervised method to identify social bots. Zhao et al [30] present a semi-supervised model founded on a attention mechanism-based graph CNN, which spots spam bots by integrating many user characteristics and relational structures. To detect counterfeit accounts from a vast volume of Twitter data, BalaAnand et al [31] presented an enhanced graph-based semi-supervised learning algorithm (EGSLA). Another work of Shaabani et al [32] present a semi-supervised self-training architecture capable of capturing Pathogenic Social Media users. To identify single and batches of spam accounts, Alharthy et al [33] use two semi-supervised techniques plus a set of specified features. A recent work of Guo [34] symmetrically involved BERT and GCN (Graph Convolutional Network, GCN),and a new architecture for bot identification that merged large-scale pre-training and transductive learning was proposed.

Numerous studies have considered the bot detection problem as a binary classification. However, only binary classifiers will be capable to differentiate bots and genuine users when bots are of the identical category as the ones used when training the model. To detect the bots, Rodriguez et al [35] used a one-class classification strategy. This strategy has the advantage of not necessitating examples of anomalous activity. When the goal is to detect deviations from predicted behavior, one-class categorization is usually applied. The researchers select the account features (retweet, replies, inter-time, number of listed tweets, and friends-to-follower ratio) and illustrated that the one-class classifier distinguishes the bots and the legitimate users consistently. Building on this idea, we suggest a one-class classification approach to extract the linguistic features that can effectively separate bots and legitimate accounts. Moreover, the proposed Hybrid-MELAu gives significant control of how to model latent spaces.

Eventually, the previous semi-supervised techniques are summarized and briefly compared in Table 1

Table 1: Brief description of prior surveyed semi-supervised methodes for bot detection.

Methode	Dataset used	Features selected	Accuracy	Precision	Recall	F1-score
Zhao et al[30]	Twitter 1KS-10KN dataset	user and network features	_	0.93	0.88	0.91
Guo [34]	cresci-rtbust, botometer-feedbak, gilani, cresci-stock-2018 and midterm dataset	tweet text	0.9026 (The best result achieved on midterm dataset)	0.8842 (The best result achieved on cresci-rtbust dataset)	0.7884 (The best result achieved on midterm dataset)	0.8089 (The best result achieved on midterm dataset)
BalaAnand et al[31]	automated data collection by Python web-scraping	Fraction of retweets, Standard tweet length, Fraction of URLs, Average time between tweets	0.903	0.923	0.908	_
Rodriguez et al[35]	Cresci-2017 dataset	account features	0.921	_	_	_
Shaabani et al [32]	ISIS dataset	_	0.82	0.90	_	_
Alharthy et al [33]	automated data collection by Twitter API	Tweet metadata and Account metadata	0.91	0.88	_	_

4 The features extraction study

4.1 Lexical, syntactic and semantic features extraction

We will examine the text content in the first part of this project to delineate bot behavior, as the language and phrase composition of bots and genuine people may differ. Although the techniques of Natural Language Processing have a redoubtable function in ensuring that bots grasp the language and more human-like, it seems that the bots be surrounded by dissension due to their restrictions to communicate with people who speak the same language [36]. To find insights into this issue, we focus on three NLP steps:

- The first step is lexical analysis. The batch of sentences and words is a language lexicon. We will first analyze the text and separate it into sentences and words. Every word and punctuation mark is a separate unit.

- The second phase is syntactic analysis. We will explore the grammatical role of every word in a sentence by tagging each of it to indicate what type of token it is, for example, is a verb (in past, present, or future tense), a pronoun, an article, a stop word, adjective . . . , and identifies the words relationship.

- In the third step, we perform the semantic analysis. To do this, we have to deploy the Word embedding techniques that mainly take words or phrases from the vocabulary to map them to real number vectors. There are many word embedding initiatives. For example, Word2vec was created for computing continuous words vector representations from huge data sets [37], 2) The GloVe stands for Word Representation Global Vectors [38]. Glove model is a log bilinear model where the possibility of the next word is calculated when the previous words are given which means the word appearances statistics in a corpus is the main source of information obtainable to all unsupervised approaches for learning word representations.

Machine learning algorithms can be used in these NLP phases to dynamically learn the rules by exploring a corpus. We start our approach by extracting features based on the previous three main analysis levels. The first process phase is the lexical analysis:

1. Divide tweets into sentences and words.
2. Elicit emojis, hashtags, both emoticons happy and emoticons sad.

3. Identify upper letters, numeric and blank spaces.
4. Then, we calculate all these features number, besides determining the whole number of characters and the average-word in both human and bots tweets.

The next step is to analyze the tweets words in terms of syntax where an ensemble of syntactic features was extracted:

1. The frequency of punctuations (commas, question mark and exclamation mark).
2. The frequency of stop words and URLs.
3. We also focus in identifying the grammatical role of each word in a sentence via speech tagging.

For the semantic approach, we realize it based on the GloVe embedding technique.

4.2 Features extraction based on lexical diversity

We study a key linguistic feature called "Lexical Diversity" (LD), which aims to indicate the complexity and the difficulty to read a text. There are many LD measures and the type-token ratio (TTR) [39, 40] is the most popular of them. It is a quantitative relation between the unparalleled words number (V) in a text and the total number of items (N) [41].

$$TTR = V/N \tag{1}$$

We take these tweets as an example: "To live with untreated PTSD is to feel as if you might die any moment. Again and again. Help cost money."
The token size in this sentence is 20 contains 18 types (to, live, with, untreated, PTSD, is, feel, like, you, might, die, any, moment, again, and, help, costs, money). In this example, the TTR is 0.90 (i.e., 18/20).

Moreover, numerous researches have demonstrated that TTR strongly relies on text length [42, 43]. The TTR value gradually decreases as the text becomes longer [44]. Consequently, some conversions of TTR raw have been suggested, this to relieve or avert this text length subordination.

The Measure of Textual Lexical Diversity (MTLD) [45, 46] creates factors from the textual sample based on the TTR values. Every factor closes when it accomplishes 0.72, which often known as the default TTR size value, and its tokens number is greater or equal to 10 tokens. Finally, the whole TTRs mean is calculated. The final MTLD result is the number of words (N) split by the number of factors reached 0.72 of TTR [41].

$$MTLD = N/factors \tag{2}$$

Then, the same process will be repeated after reversing the text and the final MTLD appreciation is calculated by averaging the two obtained MTLD values.

4.3 Features selection

In machine learning models training phase, the data characteristics get a big influence on its attainments. A bad choose of these features can injuriously influence model performance and decrease accuracy. There are many advantages of applying feature selection before shaping the data such as reducing overfitting, improving accuracy, reducing algorithm complexity, and algorithms' training faster.

For selecting features task, we used Extremely Randomized Trees Classifier(Extra Trees Classifier) [47] which is an updated Tree-Based Classifiers that extracts the most relevant features. It attach a score for every feature; if this score is high it indicates that this feature is pertinent for the model performance.

5 The proposed architecture

We introduce Hybrid-MELAu: novel semi-supervised deep framework oriented mixing engineered linguistic features based on autoencoder to improve the Twitter bot detection performance. Thus, we implement a feature extractor based on DALS (Deep Dense Autoencoder based on Lexical and Syntactic features) and GloVe-BiLSTM autoencoder (GloVe Word Embedding Bidirectional-LSTM Autoencoder) to learn better latent representations of the human linguistic behaviors. Also, there is a growing interest in bot detection to utilize one-class classifiers based solely on examples from a single class to learn its representations and determine whether a new example belongs to that class or not. Hence, the proposed architecture includes two parts: The feature learner, which relies on two autoencoders components [48, 49, 50] that pre-train the layers of the model. The feature learner associates the extracted lexical and syntactic features with their corresponding semantic features using the transfer learning technique. It consists in building the latent spaces from the pre-trained encoder part of the two autoencoders. These latent spaces present the most robust representation of the dataset. Whereas, we need to hold the concatenated encoders of the two autoencoders and fix their weights, to gain the advantage of their experience. Hence, the weights values of the encoders are frozen while we learn feed-forward deep learning network weights, following the architecture illustrated in Figure 1. Then the second part is the classification model, where we chain the features, that have been extracted, with several deep neural networks classifiers.

5.1 Features learner

To perform extracting features process, we used one of the well-known deep Representation Learning Algorithms; Autoencoders. It's a form of feedforward neural network that trains itself to match input and output.
Let's assume that only a group of unlabeled training examples exist: $\{x_1, x_2, x_3, ...\}$, where $x_i \in \Re^n$. The autoencoder compresses the input into a lower dimension then

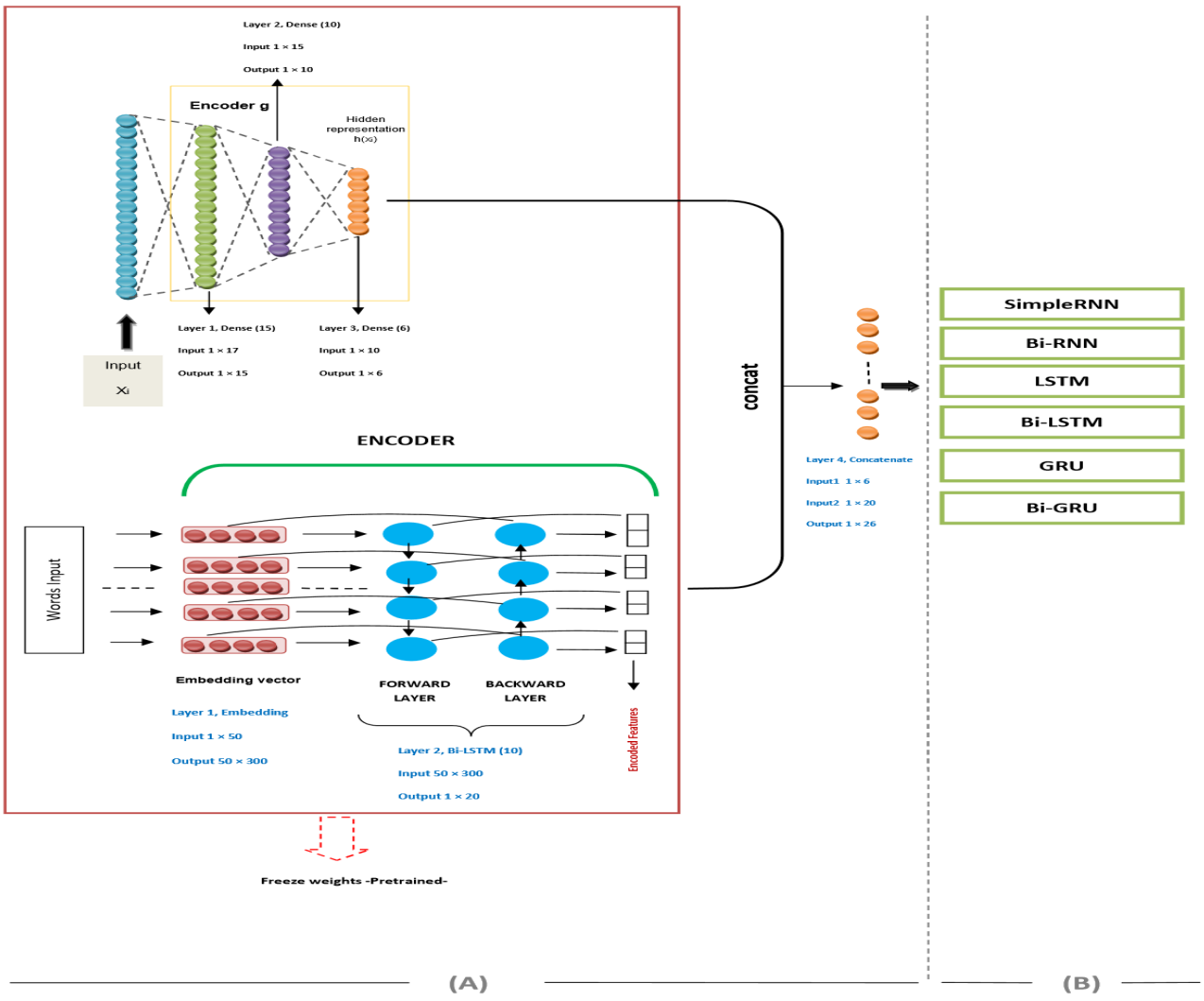

Figure 1: The Hybrid-MELAu for Twitter bots detection. Part (A) shows the freezing weights process for the two pre-trained encoders parts form DALS and Glove-BiLSTM autoencoder and their concatenation. Part (B) exhibits the classifiers.

reconstructs the output from this representation. It's an unsupervised algorithm that employs backpropagation for matching both the target and the input value: $y_i = x_i$.

The autoencoder's network is consisting of two sections: The encoder, which encodes the inputs into a hidden representation (h) or a latent space, that is capturing everything that must reconstruct the original input.

$$h = g((w * x_i) + b) \tag{3}$$

From the latent space (h), the decoder extracts the input again.

$$\widehat{x_i} = f((w' * h) + c) \tag{4}$$

Autoencoders are constrained to only copy but rather to construct and deconstruct the input. Because it's constrained by this reduction, it is forced to make priorities on which features of the input to learn, which are very useful to discriminate the humans and bots' writing style. The proposed approach comprises of two different autoencoders: the first one is based on a deep-stacked autoencoder that reconstructs the input features, and the second one is a sequence-to-sequence LSTM autoencoder that learns vector representations of any unstructured text.

The first autoencoder DALS is composed of six layers to input the lexical and syntactic features. Its architecture is provided in Figure 2.The first three layers are setting up the encoder with 15, 10, and 6 neurons respectively, while the third layer is the latent space. The last three layers perform the decoder such as the initial two layers have 10 and 15 neurons successively, and the last layer is the output layer (it outputs the same neurons numbers as the inputs). The training procedure of this architecture is summarized in Algorithm 1.

At the semantic level, the sequence prediction remains a

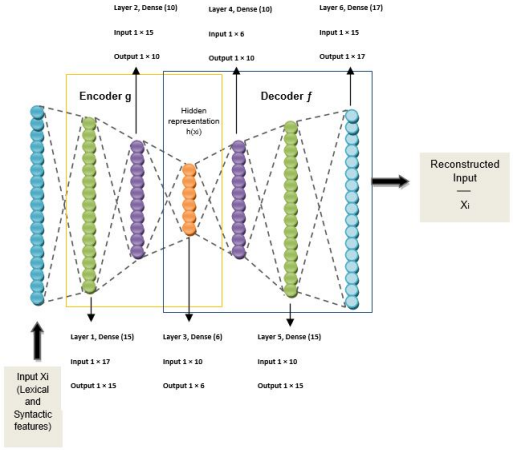

Figure 2: DALS Architecture

Algorithm 1: Deep Dense Autoencoder based on Lexical and Syntactic content.

Input : X: vector of unlabeled features
λ: hyper-parameters
T: the maximum number of iteration
Output: $\hat{X}$: reconstructed representation of the input
begin
`// Preparing data to be passed to the network`
Set t to 1
Initialize w, w', b, c
repeat
Encode the input X into the latent space h according to the equation.3.
Decompress the original input from the latent space h via the equation.4.
$E(X, \hat{X}) = ||X - \hat{X}||^2$ (the error rate).
$t = t + 1$.
Update (w, w', b, c).
until $t > T$;
return $\hat{X}$;
end

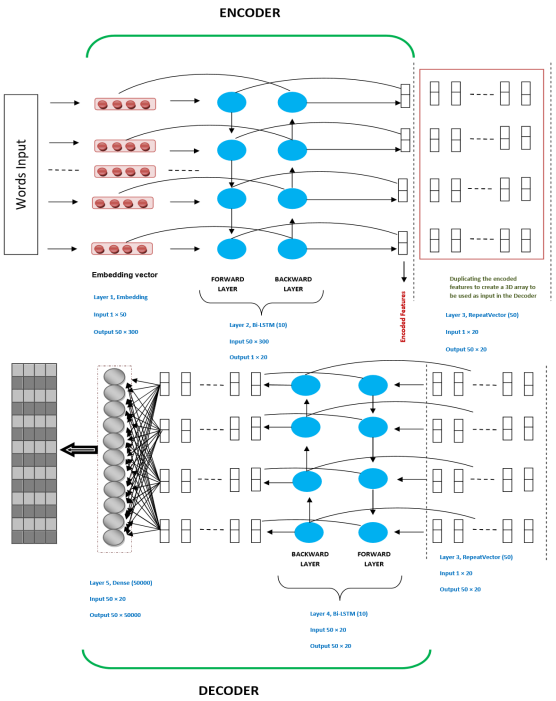

Figure 3: GloVe-BiLSTM autoencoder Flow Diagram

complex issue, not only because the input sequence length can vary, but the notes temporal scheduling can make it difficult to extract the appropriate features for use as input to supervised learning models. To capture the temporal structure, we develop a GloVe-BiLSTM autoencoder model. In another word, the encoder part of the model can be used to compress tweets text that in turn may be used as a feature vector input to a supervised learning model. For a better understanding, let's visualize the architecture in Figure 3. This figure shows the tweets flow across the GloVe-BiLSTM autoencoder network layers for one ensample of data. The encoder is accountable for the source tweet reading and encoding it to an inner representation by capturing the meaning of these tweets. A simple model creation includes an embedding input ensued by a Bidirectional-LSTM hidden layer that generates a fixed-length representation. First, we input the tweet texts to the embedding layer, where each word is transformed into a distributed representation [51]. This layer is a matrix of size $m \times v$, where v is the vector length and it's equal to 300, in which we learned word embeddings from text using a pre-trained 300-dimensional Google News Vectors approach (GloVe) [38], and m is the tokens number in the tweets which is fixed on 50.

The Bidirectional-LSTM layer [52] used the hidden states to maintain the inputs information and fed it in a forward way from past to future and backward from future to past. Moreover, Bidirectional LSTMs have the capacity to better understand the context [53]. After that, we add the decoder which is a Bidirectional-LSTM layer. It assumes a three dimensional input for creating a decoded sequence of various lengths determined by the problem. So we configure first the RepeatVector layer to create a three dimensional BiLSTM output. Then, like the encoder, one Bidirectional-LSTM layer with the same number of cells was utilized in the decoder model implementation. Finally, the dense layer generates the autoencoder output which is also a matrix of size $m \times n$ which n is the tweet corpus (50.000).

The training procedure of the GloVe-BiLSTM autoencoder is summarized in Algorithm 2.

Algorithm 2: GloVe-BiLSTM autoencoder

Input : S: set of tweets
K: set of tokens in one tweet: size m
C: The tweet corpus: size n
Batch: the number of training examples utilized in one iteration: size z
θ: hyper-parameters
Output: $\hat{P}$: reconstructed matrix: size $(m \times n)$
begin
// Preparing data to be passed to the stack
foreach $s \in S$ **do**
$s \leftarrow$ nlp.prepossessing (s)
end
repeat
foreach *Batch* **do**
// Calculating embeddings for each token
foreach $k \in K$ **do**
emb (k) $\leftarrow$ glove (k)
end
Encoder = Build-Model(LSTM_Bidirectional.input ($[m \times Embedding\_size], \theta$))
Encoder_output $\leftarrow [1 \times The double number of cells]$
// repeat Encoder_output m times to create 3D vector
repeat (*Encoder_output,*m)
output $\leftarrow [m \times$Number of cells$*2]$
Decoder = Build-Model(LSTM_Bidirectional.input ($output, \theta$))
Decoder_output $\leftarrow [m \times$ Number of cells$*2]$
// Generating the output using fully connected layer with size n
$\hat{P} = [m \times n]$
end
until *Untill convergence*;
return $\hat{P}$;
end

5.2 Predictor model

The key idea of our proposed framework (Hybrid-MELAu) is using transfer learning [54], by copying both the pre-trained encoder part of DALS and GloVe—BiLSTM first n layers to the n first layers of the deep learning classifiers. The implemented classifiers are 1)—a Recurrent Neural Network (RNN) [55] classifier which is a universal approximation of dynamical systems, 2)—a Long short-term memory networks (LSTMs) [56] predictor which considered as an update of RNN that used on several works for example in Kalyoncu [57] research, 3)—a Gated recurrent units (GRU) [58] classifier, 4)- a three bidirectional architectures (BiRNN[59],BiLSTM and BiGRU [60]).
During the predictor models training, we set the mean squared error (MSE) as a cost function. It is defined below:

$$L(Y, f(X, s)) = L(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^{N} (Y, \hat{Y})^2 \qquad (5)$$

Where N is the feature dimensionality, X is the features vectors and s is a set of tweets, Y is the output ground truth, and $\hat{Y}$ is the predicted output (Human or Bot).

Using a pre-trained network that is trained on data with one class only ensures that the bot detection task is performed based on the most frequent characteristics of non-intrusion samples.

6 Experiments results

6.1 Dataset

Several tweeter real-world datasets are used in our research. The first is defined in [61, 62]. According to [61]), genuine accounts consists of 3,474 real users accounts with $8,377,522$ tweets. The bots accounts separated on three datasets. During the 2014 Romanian Mayoral election, the social spambots1 dataset was scraped from Twitter, it is composed of 991 accounts and $1,610,176$ tweets. Spambots 2 dataset is a group of $3,457$ bots accounts who passes many months promulgating the #TALNTS hashtag through $428,542$ tweets. Where this last concerns a mobile phone application for contacting and recruiting artists working in several fields. The immense generality of tweets were innocuous statements, sporadically scattered by tweets naming a specific human account and recommending that he purchase the VIP edition of the software from a Web store. The dataset of Spambots 3 is a set of 464 accounts and $1,418,626$, this dataset announced products for selling on Amazon.com. The delusive activity is executed by spamming URLs referring to the publicized products.

The second one is the celebrity dataset which contains celebs' accounts [63]. The Center for Complex Networks and Systems Research at Indiana University (CNetS team) collected 5,918 celebrity human accounts. We also add two other datasets: pronbots-2019 and political-bots [63]. Pronbots-2019 is a set of 21,963 bot accounts distributed by Andy Patel. Political-bots is a set of 62 Automated political accounts.

6.2 Exploratory study results

To extract syntactic and lexical characteristics, we have applied NLP analysis approaches as explicated in section 4. Hither, we consider a sample of 1 000 000 data containing an equal number of human and bots tweets and compare the writing style of both at the lexical level. The findings of this comparison are illustrated in Figure.4. We observe that

humans use a greater number of hashtags than bots. Also, they use different number of emotions type, numbers, sentences, words, blank space, and upper letters. It is due to the fact that the human can easily diversify their lexical context.

Then, we compare the syntactic features analysis results based on URLs, punctuation, and stop words. As we can see in Figure.5, the number of different syntactic tokens could be different, especially since a successful bot can use a linguistic approach based on the linguistic structure. For syntactic analysis based on speech-tagging, there are many tags, so, we have just focused on recognizing some tags, that essentially help in the interpretation of the given sentence. Besides, we compare the different POS tagging features in the bots and human writing styles (Figure.6).

From Figure 6, we notice that bots used to write their tweets, the following features: proper plural noun, proper singular noun, plural noun, singular noun, prepositions, coordinating conjunctions, determiners, modal, verbs 3sg pres, verbs base form, adjective, comparative adjective, superlative adjective, superlative adverbs and comparative adverbs much more than humans. Because these characteristics are considered as basic units (tokens) in the construction of the sentences and aren't difficult to simulate. Although this is a good bot imitation, they haven't been able to outperform humans in terms of features shown on the right side (personal pronoun, adverb, verb past tense, verb present participle, verb past participle, verb non-3sg pres, interjections, and foreign words from other languages). The key idea is that exploiting this feature set is more complicated and requires special conditions. For example, humans make the use of various interjections with rich context. Therefore, we can conclude that human vary their tone in writing depending on their feelings, the reader, and the events by using empathy, encouragement and astonishing events.

For the lexical richness task, We chose the MTLD metric due to the fact it is a robust lexical diversity indicator that is unaffected by sample length. [64].
First, we compute all the POS (part-of-speech) tagged to rich inflectional languages. After that, we compute the MTLD. Figure.7 shows how the MTLD metric varies between the human and bot tweets.
As we can observe from Figure 7 and Table 2, the range of bot's MTLD values is bigger than the human ranges values. The maximum value of the bots' MTLD is higher than the humans' MTLD while both minimum values are equals. It can be seen that MTLD is a metric of analyzing the number of consecutive words supported by a specific type-token ratio. We observe that a well automated bot rely on the NLP rules to generate a rich lexicon but human are using an odd approach as their writing skills outperform a simple NLP rules.

6.3 Hybrid-MELAu evaluation results

For this evaluation phase, we have selected the 17 highest linguistic features that have a great impact on predictability (see Figure.8). We split Cresci datasets into approximately 80% training and 20% testing set. As mentioned in the 4.2 subsection, the two autoencoders will be trained on data with one class only to ensure that the prediction task is performed based on the most frequent characteristics of human samples. So, the training group is divided again based on the dataset label (Human and Bot). The human and bot label rates are respectively 54% and 46% of the training set. Then, we rely on the training set of the human class. After dividing dataset into 75% training set and 25% validation set, and for retrieving the best hyper-parameters of the two autoencoders, we used one of the optimization approaches that are provided in scikit-learn: "GridSearchCV" [65]. It evaluates all potential values of parameter composition and retains the best one. Table 3 shows the autoencoders hyper-parameters after using GridsearchCV.
The top hyper-parameters are:

- Utilization 256 as a batch size for the both autoencoders.
- The usage of "Adam" and "Nadam" separately as optimizer functions for the DALS and GloVe-BiLSTM autoencoder.
- The DALS loss function is MSE and for GloVe-BiLSTM autoencoder is sparse-categorical-crossentropy.
- The learning rate values for the first autoencoder and the second one are 0.0001 and 0.001.
- For the hidden activation function the choice fell on "relu" for the DALS and "tanh" for the GloVe-BiLSTM autoencoder. And for the output activation function, linear and softmax functions were selected respectively for the two autoencoders.

First, the two autoencoders were trained on the dataset based on human class only using the selected linguistic features and the best hyper-parameters. Then, the two encoder parts are frozen and cemented to make one features vector. After this phase, six recurrent neural networks models were built as follows: the feature vector was repeated once to create a 3D output utilizing RepeatVector, and it's fed to the next layer of six classifiers: (1 — SimpleRNN classifier, 2 — BiRNN classifier, 3 — LSTM classifier, 4 — BiLSTM classifier, 5 — GRU classifier, 6 — BiGRU classifier). The units number of cells in each one is fixed at 300. Afterward, to make a one-dimensional vector a flatten layer was added. For rending the model more powerful, the output vector is passed to a fully connected layer. Then, the last layer transforms its input into a one result using the sigmoid function [66].

The different recurrent neural classifiers are implemented on the whole labeled dataset with 55% of data for

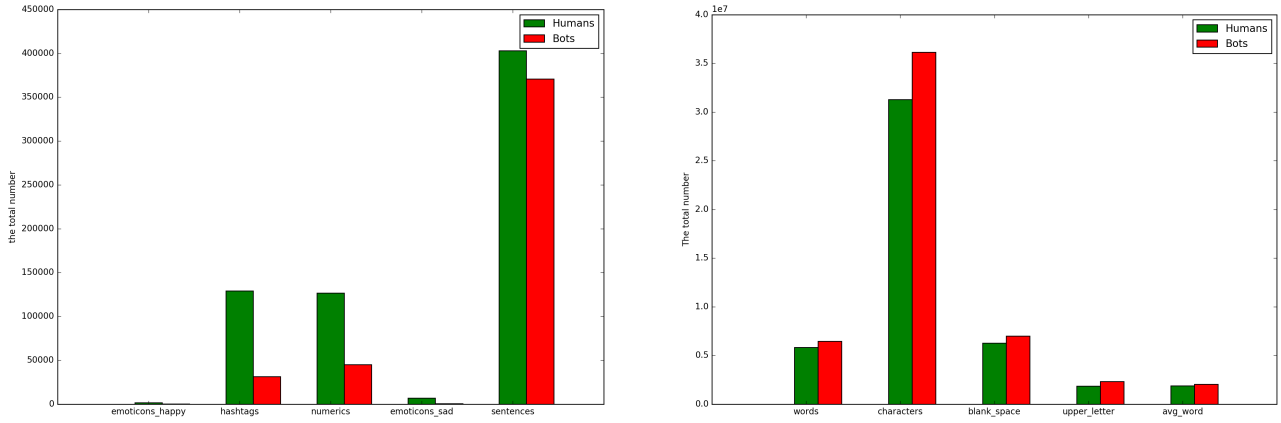

Figure 4: Comparison between the writing style of both human and bots at the Lexical level.

Table 2: Summary values of Measure of Textual Lexical Diversity distribution across the two lables.

Measure of Textual Lexical Diversity (MTLD)							
	Min	Max	Mean	25th percentile	Median	75th percentile	IQR
Human	1.0	69.91	25.23	6.0	11.0	31.5	25.5
Bot	1.0	81.55	27.38	8.0	14.0	37.33	29.33

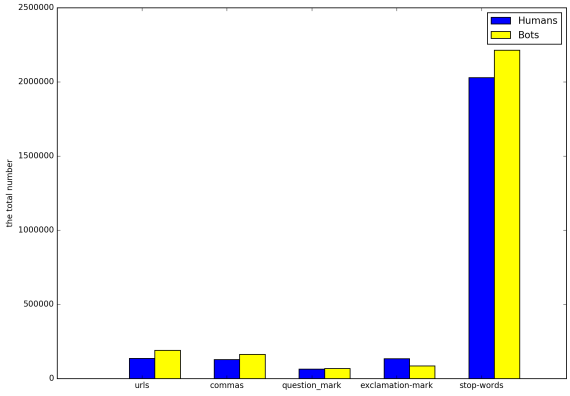

Figure 5: Comparison between the writing style of both human and bots at the syntactic level.

the training set, the previous preserved testing set (20% of data), and 25% of data for the validation set using Google Colab environment. The runtime had configured to use Keras [67] API v2.4.3, Tensorflow v2.4.0, Python 3.6.9 - 64bit-, a GPU Hardware accelerator. The classifiers were trained for 150 epochs with 256 as a batch size using Adam as optimization function and mse as loss function. For the fully connected layer and output layer we used ReLu and segmoid activation function respectively. We employ different metrics : *Precision*, *Recall*, *F-Measure*, *Accuracy* and *Matthew Correlation Coefficient (MCC) [68]* to compare the classifiers performance.

Experiments on the Cresci dataset show that it is possible to forecast with a high degree of accuracy. As we can see from Table 4, the Hybrid-MELAu+BiRNN classifier shows high performance for bot detection, and it is better than the other recurrent classifiers when the overall accuracy is 92.22%. All recurrent classifiers had closely comparable performance.

After that, because our Hybrid-MELAu (with BiRNN) model falls under the semi-supervised techniques, we choose to compare its performance with the methods mentioned in Table 1, the results are presented in Figure 9.

As we can see from Figure.9, the Hybrid-MELAu outperformed the other models in terms of the different metrics. In fact, in this work, we emphasize linguistic features without taking into account the users' features to discriminate the human's and bots writing style behavior. Hence, this result illustrates the ability of the feature learner based on autoencoders with transfer learning to generate elite features from latent spaces from the pre-trained encoder part.

Certainly that the linguistic features capture sentence level and word level complexity using different lexical and syntactic indexes influence bot identification and show better results. It also showed that, when compared to human accounts, Bot accounts have a high non-homogenize in their discriminatory behavioral characteristics [25], designing a deep linguistic framework with transfer learning founded only on collections of linguistic characteristics is able to define if a single tweet is being written by a human or a bot with good accuracy. Moreover, the generative ability of the part of the pre-trained encoder enhances the predictor to discern differences in the writing styles of both humans and bots.

7 Discussion

In this section, we will discuss the main findings of the manuscript and address its implications. Moreover, we will

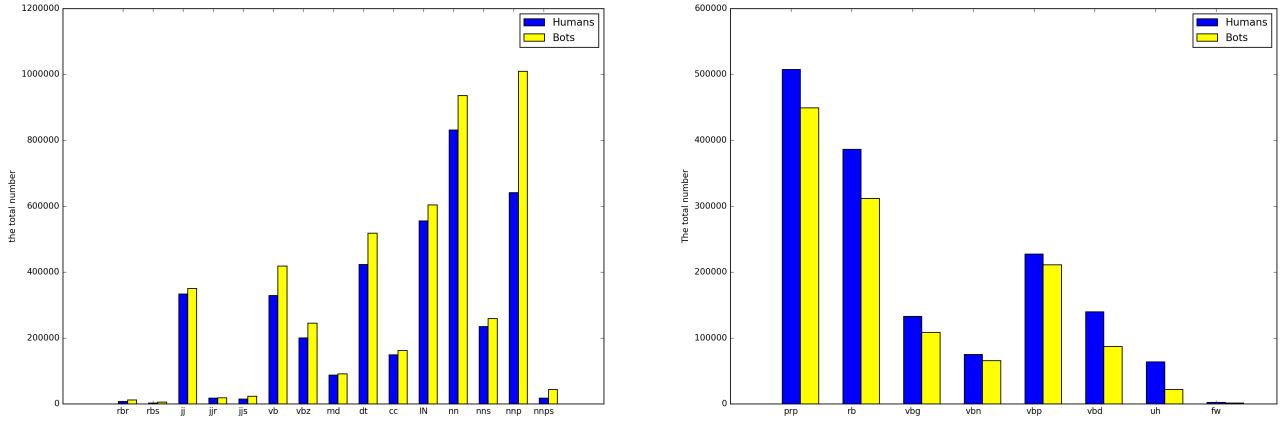

Figure 6: Comparison of different POS tagging features. the left part finds out the most characteristics employed by bots in comparison to humans, while the right side shows the features for which humans surpassed the bots.

Table 3: GridsearchCV for the best hyper-parameters optimization.

	Optimizer	Hidden Activation Function	Output Activation Function	Loss	batch size	learning rate	
1	SGD	softmax	softmax	mse	16	0.00001	
2	RMSprop	softplus	softplus	sparse-categorical-crossentropy	32	0.0001	
3	Adagrad	softsign	softsign	msle	64	0.001	
4	Adadelta	relu	relu	categorical-crossentropy	128	0.01	
5	Adam	tanh	tanh	kullback-leibler-divergence	256	0.1	
6	Adamax	sigmoid	sigmoid	mae	512	-	
7	Nadam	hard-sigmoid	hard-sigmoid	binary-crossentropy	-	-	
8	-	linear	linear	hinge	-	-	
9	-	elu	elu	squared-hinge	-	-	
10	-	selu	selu	-	-	-	
DALS	**Adam**	**relu**	**linear**	**mse**	**256**	**0.0001**	
GloVe-BiLSTM autoencoder	**Nadam**	**tanh**	**softmax**	**sparse-categorical-crossentropy**	**256**	**0.001**	

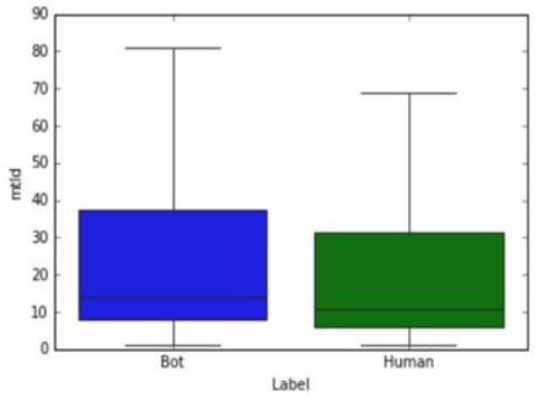

Figure 7: Variation of MTLD metric between the human and bot tweets

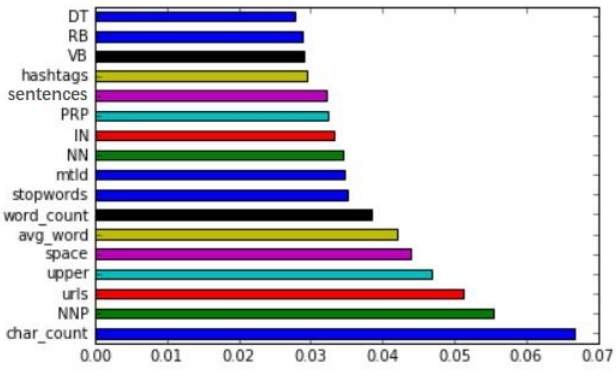

Figure 8: Top 17 most important features in the data using Extra Trees Classifier

Table 4: Comparison among the various presented approaches in terms of performance.

	Precision	**Recall**	**F1-score**	**Accuracy**	**Loss**	**MCC**
Classifiers:						
Hybrid-MELAu+SimpleRNN	0.92455	0.9102	0.91375	0.9154	0.0718	0.8347
Hybrid-MELAu+BiRNN	**0.9318**	**0.9169**	**0.92065**	**0.9222**	**0.0654**	**0.8486**
Hybrid-MELAu+LSTM	0.9231	0.908	0.91165	0.9134	0.0728	0.8310
Hybrid-MELAu+BiLSTM	0.93085	0.9168	0.92035	0.9219	0.0657	0.8476
Hybrid-MELAu+GRU	0.92195	0.90705	0.91065	0.9124	0.0740	0.8289
Hybrid-MELAu+BiGRU	0.9311	0.9166	0.92025	0.9218	0.0658	0.8476

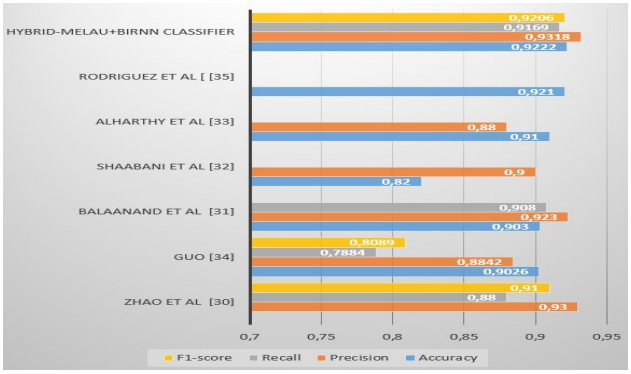

Figure 9: Experiments Results.

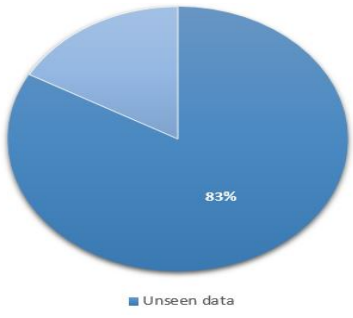

Figure 10: The prediction accuracy of Hybrid-MELAU+BiRNN classifier on an unseen data (Celebrity, pronbots-2019 and political bots dataset)

test the proposed model robustness by introducing further experiment. Whilst the majority of work on bot detection has focused on investigating various sets of features, our first concern in this present research is the analysis of the bot writing style through using the Natural Language Processing (NLP) to find insights about how the linguistic features helps in bots detection. Our research reveals that certain lexical and syntactic measures are the most significant signs that contribute to distinguishing the writing style of both bots and humans. In fact, the exploratory analysis showed that humans could infer the relationship between different contexts by employing a context-related lexical level (as discussed in section 6.2). Unlike bots, humans intend to express and argue their ideas using numeric (digit, date, real numbers). In addition, humans use more phrases in one tweet than the bots.

According to the syntactic analysis based on speech-tagging (see Figure 6), we find that although humans master the language's syntax, they show creative behavior in their writing style. Therefore, humans make the use of interjections in a specific sentence related to their psychological state and their feelings. They might also express a position whether the latter is personal or related to another person. For example, in this human tweet "hmm fishy!!" an exclamation sentence existed, conveying that the person expresses arousing feelings of doubt or suspicion. As we can note there is no grammatical structure in this sentence, it's just composed of two words, an interjection (hmm) and an adverb (fishy). Furthermore, the human explains a specific statement taking profit from a variety of adverbs and personal pronouns. It means that they tend to diversify their writing styles according to a somewhat odd approach to tweeting their ideas, such as using words in foreign languages. They don't also focus on one tense to conjugate verbs. These results represent a strong conclusion to discern the difference between humans' and bots' writing styles.

Furthermore, computing the lexical diversity measures (see Figure.7) would further disseminate the writing style from humans and bots, which can be seen differently in a text depending on specific type-token ratio and vocabulary knowledge. We conclude that a successful (well automated) bot includes the NLP approaches to generate tweets and get a rich lexicon. Meanwhile, the human includes their skills with the language to write in an intelligent way

("To live with untreated PTSD is to feel like you might die any moment. Again and again. Help costs money."). Despite the fact that the machine learning techniques used in bots through NLP have improved their ability to generate content with high lexical diversity, as we can see from this bot tweets: "Today's Inspirational Quote Climb the mountains and get their good tidings. Nature's peace will flow into you as...", there is still a lot to do to imitate the smart human writing.

Our second concern was how to develop a hybrid deep learning approach based only on linguistic features that can improve the detection performance. This can be achieved by building a framework in a hybrid fashion Mixing Engineered Linguistic features based on Autoencoders (Hybrid-MELAu). In fact, deep neural networks' versatility allows them to integrate numerous neural building blocks to construct a more powerful hybrid model by complementing one another. The autoencoder has shown to be a useful model for modeling latent distributions since it allows you a lot of control.

To demonstrate the model's sturdiness, we tested our framework's prediction performance on a new unseen dataset that combined three datasets: celebrity, pronbots-2019, and political bots. The capacity of a predictive model to perform well over a variety of data sets determines its robustness. Therefore, the resilience of the models built in this study was tested on this new dataset after they had been trained with the Cresci dataset. Figure 10 illustrates a good prediction result when applied to an unseen dataset. Our framework ensures efficient detection because once the autoencoder model is trained, its results will be used directly for transfer learning without the need to resort to the two features of learners' training. The fact that our framework is semi-supervised with one-class authorize benefits from the myriad of unlabeled training data for learning task performance amelioration because the amount of unlabeled samples is generally greater and more accessible than the number of labeled samples. Finally, the findings of this work also show that pre-trained models based on transfer learning are able to improve the accuracy of the bots detection. Surprisingly, a set of linguistic features, such as those obtained from our exploratory study, are effective in distinguishing social bots. In future works, since we have found that the linguistic deep framework with transfer learning model is discernible of the bots writing style, we are going to incorporate the different set of features in our framework. This could help for social bots detection accuracy improving.

8 Conclusion

We develop the Hybrid-MELAu: a semi-supervised framework to model different mixing engineered linguistic features based on autoencoder, and use the transfer learning to take profit from its strong ability to generalize to unseen samples, which improve the social bots detection. The framework is composed of two essential parts: the features learner and the predictor. The features learner combine two encoder part from the following two components: i) the DALS and ii) The GloVe-BiLSTM. The DALS maps the content features to higher-order features, which enables the lexical richness to be encompassed. The GloVe-BiLSTM trains two LSTMs instead of one on the input sequences. This can provide reliable semantic features and result in accurate learning on the detection. The proposed approach captures different lexical and syntactic indexes that influence bot detection and shows significant results. Our new mechanism for detecting bot based on a mining writing style effectively detects bots with a 92.22% accuracy rate.

Finally, to confirm the gained results and implement a more until study, we plan to apply our approach to datasets with long corpus, length to provide deep insights about the text diversity impact on the detection process. Furthermore, highlighting human behavioral trends might be a fruitful direction for future research, such as their activity and their dynamics, which can be associated with linguistic features.

References

[1] E. Kajan, N. Faci, Z. Maamar, M. Sellami, E. Ugljanin, H. Kheddouci, D. Stojanovic, and D. Benslimane, "Real-time tracking and mining of users' actions over social media," *Computer Science and Information Systems*, vol. 17, pp. 403–426, 2020. [Online]. Available: https://doi.org/10.2298/CSIS190822002K

[2] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, 2020. [Online]. Available: https://doi.org/10.1145/3395046

[3] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *SIGKDD Explor. Newsl.*, vol. 19, no. 1, p. 22–36, 2017. [Online]. Available: https://doi.org/10.1145/3137597.3137600

[4] B. M. Amine, A. Drif, and S. Giordano, "Merging deep learning model for fake news detection," in *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, 2019, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICAEE47123.2019.9015097

[5] L. Azevedo, M. d'Aquin, B. Davis, and M. Zarrouk, "Lux (linguistic aspects under examination): Discourse analysis for automatic fake news classification," in *ACL/IJCNLP (Findings)*, 2021, pp. 41–56. [Online]. Available: https://doi.org/10.18653/v1/2021.findings-acl.4

[6] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari,

M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, and T. Mandl, "The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news," in *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 639–649. [Online]. Available: https://doi.org/10.1007/978-3-030-72240-1_75"

[7] Z. Ferhat Hamida, A. Refoufi, and A. Drif, "Fake news detection methods: A survey and new perspectives," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)*. Cham: Springer International Publishing, 2022, pp. 123–141. [Online]. Available: https://doi.org/10.1007/978-3-030-90639-9_11

[8] D. Kosmajac and V. Keselj, "Twitter bot detection using diversity measures," in *Proceedings of the 3rd International Conference on Natural Language and Speech Processing*. Trento, Italy: Association for Computational Linguistics, 2019, pp. 1–8.

[9] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2019, pp. 101–109. [Online]. Available: https://doi.org/10.1109/TPS-ISA48467.2019.00021

[10] R. De Nicola, M. Petrocchi, and M. Pratelli, "On the efficacy of old features for the detection of new bots," *Information Processing Management*, vol. 58, no. 6, p. 102685, 2021. [Online]. Available: https://doi.org/10.1016/j.ipm.2021.102685

[11] I. Alsmadi and M. J. O'Brien, "How many bots in russian troll tweets?" *Information Processing Management*, vol. 57, no. 6, p. 102303, 2020. [Online]. Available: https://doi.org/10.1016/j.ipm.2020.102303

[12] N. El-Mawass, P. Honeine, and L. Vercouter, "SimilCatch: Enhanced social spammers detection on Twitter using Markov Random Fields," *Information processing management*, vol. 57, p. 102317, 2020. [Online]. Available: https://doi.org/10.1016/j.ipm.2020.102317

[13] K.-C. Yang, P.-M. Hui, and F. Menczer, "How twitter data sampling biases u.s. voter behavior characterizations," *ArXiv*, vol. abs/2006.01447, 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2006.01447

[14] K. C. Yang, P. M. Hui, and F. Menczer, "Bot electioneering volume: Visualizing social bot activity during elections," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 214–217. [Online]. Available: https://doi.org/10.1145%2F3308560.3316499

[15] P. Pham, L. T. Nguyen, B. Vo, and U. Yun, "Bot2vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks," *Information Systems*, vol. 103, p. 101771, 2022. [Online]. Available: https://doi.org/10.1016/j.is.2021.101771

[16] L. Nizzoli, S. Tardelli, M. Avvenuti, S. Cresci, M. Tesconi, and E. Ferrara, "Charting the landscape of online cryptocurrency manipulation," *IEEE Access*, vol. 8, pp. 113 230–113 245, 2020. [Online]. Available: https://doi.org/10.1109%2Faccess.2020.3003370

[17] F. Giglietto, N. Righetti, L. Rossi, and G. Marino, "Coordinated link sharing behavior as a signal to surface sources of problematic information on facebook," in *International Conference on Social Media and Society*, ser. SMSociety'20. New York, NY, USA: Association for Computing Machinery, 2020, p. 85–91. [Online]. Available: https://doi.org/10.1145/3400806.3400817

[18] L. Luceri, A. Deb, A. Badawy, and E. Ferrara, "Red bots do it better:comparative analysis of social bot partisan behavior," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1007–1012. [Online]. Available: https://doi.org/10.1145/3308560.3316735

[19] L. Luceri, F. Cardoso, and S. Giordano, "Down the bot hole: Actionable insights from a one-year analysis of bot activity on twitter," *First Monday*, vol. 26, no. 3, 2021. [Online]. Available: https://doi.org/10.5210/fm.v26i3.11441

[20] E. Ferrara, O. Varol, F. Menczer, and A. Flammini, "Detection of promoted social media campaigns," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 10, 2016, pp. 563–566.

[21] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Dna-inspired online behavioral modeling and its application to spambot detection," *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 58–64, 2016. [Online]. Available: https://doi.org/10.1109/MIS.2016.29

[22] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, pp. 312–322, 2018. [Online]. Available: https://doi.org/10.1016/j.ins.2018.08.019

[23] K. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *AAAI*, 2020. [Online]. Available: https://doi.org/10.1609/aaai.v34i01.5460

[24] M. Heidari and J. H. Jones, "Using bert to extract topic-independent sentiment features for social media bot detection," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2020, pp. 0542–0547. [Online]. Available: https://doi.org/10.1109/UEMCON51285.2020.9298158

[25] M. Sayyadiharikandeh, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "Detection of novel social bots by ensembles of specialized classifiers," *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020. [Online]. Available: http://doi.org/10.1145/3340531.3412698

[26] S. Kumar, S. Garg, Y. Vats, and A. S. Parihar, "Content based bot detection using bot language model and bert embeddings," in *2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2021, pp. 285–289. [Online]. Available: https://doi.org/10.1109/ICCCSP52374.2021.9465506

[27] V. Gaurav, S. Singh, A. Srivastava, and S. Shidnal, "Codescan: A supervised machine learning approach to open source code bot detection," in *Applied Information Processing Systems*. Singapore: Springer Singapore, 2022, pp. 381–389. [Online]. Available: https://doi.org/10.1007/978-981-16-2008-9_37

[28] A. Praveena and S. Smys, "Effective spam bot detection using glow worm-based generalized regression neural network," in *Mobile Computing and Sustainable Informatics*. Singapore: Springer Singapore, 2022, pp. 469–487. [Online]. Available: https://doi.org/10.1007/978-981-16-1866-6_34

[29] M. Chakraborty, S. Das, and R. Mamidi, "Detection of fake users in twitter using network representation and nlp," in *2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2022, pp. 754–758. [Online]. Available: https://doi.org/10.1109/COMSNETS53615.2022.9668371

[30] C. Zhao, Y. Xin, X. Li, H. Zhu, Y. Yang, and Y. Chen, "An attention-based graph neural network for spam bot detection in social networks," *Applied Sciences*, vol. 10, no. 22, 2020. [Online]. Available: https://doi.org/10.3390/app10228160

[31] B. Muthu, K. Natesapillai, K. Subburathinam, R. Varatharajan, G. Manogaran, and C. B. Sivaparthipan, "An enhanced graph-based semi-supervised learning algorithm to detect fake users on twitter," *The Journal of Supercomputing*, vol. 75, 09 2019. [Online]. Available: https://doi.org/10.1007/s11227-019-02948-w

[32] E. Shaabani, A. Sadeghi-Mobarakeh, H. Alvari, and P. Shakarian, "An end-to-end framework to identify pathogenic social media accounts on twitter," *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, pp. 128–135, 2019. [Online]. Available: https://doi.org/10.48550/arxiv.1905.01553

[33] R. Alharthy, A. Alhothali, and K. Moria, "Detecting and characterizing arab spammers campaigns in twitter," *Procedia Computer Science*, vol. 163, pp. 248–256, 01 2019. [Online]. Available: https://doi.org/10.1016/j.procs.2019.12.106

[34] 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, 2022. [Online]. Available: https://doi.org/10.3390/sym14010030

[35] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, "A one-class classification approach for bot detection on twitter," *Computers & Security*, vol. 91, p. 101715, 2020. [Online]. Available: https://doi.org/10.1016/j.cose.2020.101715

[36] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW '16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2016, p. 273–274. [Online]. Available: https://doi.org/10.1145/2872518.2889302

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. [Online]. Available: https://doi.org/10.48550/arxiv.1301.3781

[38] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/D14-1162

[39] J. W. Chotlos, "Iv. a statistical and comparative analysis of individual written language samples," *Psychological Monographs*, vol. 56, p. 75–111, 1944. [Online]. Available: https://doi.org/10.1037/h0093511

[40] M. Templin, *Certain Language Skills in Children: Their Development and Interrelationships*. Minneapolis,MN: University of Minnesota Press, 1957. [Online]. Available: https://doi.org/10.1086/459642

[41] P. Lissón and N. Ballier, "Investigating lexical progression through lexical diversity metrics in a corpus of french l3," *Discours [En ligne]*, vol. 23, 2018. [Online]. Available: https://doi.org/10.4000/discours.9950

[42] N. Chipere, D. Malvern, and B. Richards, "Using a corpus of children's writing to test a solution to the sample size problem affecting type-token ratios," in *Corpora and language learners*. John Benjamins, 2004, pp. 139–147. [Online]. Available: https://doi.org/10.1075/scl.17.10chi

[43] K. Kettunen, "Can type-token ratio be used to show morphological complexity of languages?" *Journal of Quantitative Linguistics*, vol. 21, p. 223–245, 2014. [Online]. Available: https://doi.org/10.1080/09296174.2014.911506

[44] H. Heaps, *Information Retrieval: Computational and Theoretical Aspects*. New York: Academic Press, 1978. [Online]. Available: https://doi.org/10.5860/crl_40_03_276

[45] P. M. MacCarthy, "An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity," Ph.D. dissertation, University of Memphis, 2005.

[46] P. McCarthy and S. Jarvis, "Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment," *Behavior Research Methods*, vol. 42, pp. 381–92, 2010. [Online]. Available: https://doi.org/10.3758/BRM.42.2.381

[47] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach Learn*, vol. 63, p. 3–42, 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-6226-1

[48] Y. Lecun, "Modeles connexionnistes de l'apprentissage (connectionist learning models)," Ph.D. dissertation, Universite de Paris VI, 1987.

[49] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, p. 291–294, 1988. [Online]. Available: https://doi.org/10.1007/BF00332918

[50] G. Hinton and R. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Advances in Neural Information Processing Systems*, vol. 6. Morgan-Kaufmann, 1994, pp. 3–10. [Online]. Available: https://doi.org/10.5555/2987189.2987190

[51] K. Lopyrev, "Generating news headlines with recurrent neural networks," *CoRR*, vol. abs/1512.01712, 2015. [Online]. Available: https://doi.org/10.48550/arxiv.1512.01712

[52] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 799–804. [Online]. Available: https://doi.org/10.1007/11550907_126

[53] A. Kulkarni and A. Shivananda, *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python*. Apress, 2019. [Online]. Available: https://doi.org/10.1007/978-1-4842-4267-4

[54] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 2014. [Online]. Available: https://doi.org/10.48550/arxiv.1411.1792

[55] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, p. 533–536, 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[56] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[57] S. Kalyoncu, A. Jamil, E. Karataş, J. Rasheed, and C. Djeddi, "Stock market value prediction using deep learning," *Data Science and Applications*, vol. 3, no. 2, pp. 10–14, 2020. [Online]. Available: https://doi.org/10.1186/s40537-020-00333-6

[58] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 103–111. [Online]. Available: https://doi.org/10.3115/v1/W14-4012

[59] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673–2681, 1997. [Online]. Available: https://doi.org/10.1109/78.650093

[60] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," *ArXiv*, vol. abs/1603.01417, 2016. [Online]. Available: https://doi.org/10.48550/arxiv.1603.01417

[61] 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*, ser. WWW '17

Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 963–972. [Online]. Available: https://doi.org/10.1145/3041021.3055135

[62] S. Cresci, "Mib datasets," http://mib.projects.iit.cnr.it/dataset.html, 2017, accessed: 2021-01-12.

[63] 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. [Online]. Available: https://doi.org/10.1002/hbe2.115

[64] G. Fergadiotis, H. H. Wright, and S. B. Green, "Psychometric evaluation of lexical diversity indices: Assessing length effects," *Journal of Speech, Language, and Hearing Research*, vol. 58, no. 3, pp. 840–852, 2015. [Online]. Available: https://doi.org/10.1044/2015_JSLHR-L-14-0280

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: https://doi.org/10.5555/1953048.2078195

[66] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 195–201. [Online]. Available: https://doi.org/10.1007/3-540-59497-3_175

[67] F. Chollet, "Keras: Theano-based deep learning library, 2015,," http://keras. io, 2015, accessed: 2021-01-18.

[68] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000. [Online]. Available: https://doi.org/10.1093/bioinformatics/16.5.412