**Final version of the accepted paper. Cite as:** *"Mohd Fazil, Amit Kumar Sah, and Muhammad Abulaish, A Hierarchical Attention-Based Neural Network Model for Socialbot Detection in OSN, In Proceedings of the 19th IEEE/WIC/ACM International Conference on Web Intelligence (WI), Malbourne, Australia, December 14-17, 2020, pp. 1-6."*

# A Hierarchical Attention-Based Neural Network Model for Socialbot Detection in OSN

Mohd Fazil
*Department of Computer Science*
*South Asian University, N. Delhi, India*
Email: mohdfazil.jmi@gmail.com

Amit Kumar Sah
*Department of Computer Science*
*South Asian University, N. Delhi, India*
Email: amitcsrs@students.sau.ac.in

Muhammad Abulaish
*Department of Computer Science*
*South Asian University, N. Delhi, India*
Email: abulaish@ieee.org

*Abstract*—**In this paper, we present a hierarchical attention-based deep neural network model for socialbots detection in online social networks through modeling user behavior at two levels of granularity leveraging profile, activity, temporal, and content information. The proposed approach is novel from two perspectives – (i) it models user representations using a comprehensive set of profile and behavior information, and (ii) it applies hierarchical attention mechanism at both low-level and high-level user representations. The proposed approach jointly models profile, temporal, and activity information as sequences, which are given as input to a two-layer stacked BiLSTM, while content information is given to a six-layer CNN. On evaluation, our model performs significantly better in comparison to the baselines and state-of-the-art methods.**

*Index Terms*—**Social network analysis, Socialbot detection, Hierarchical attention, Convolutional neural network, BiLSTM.**

## I. Introduction

In the 21st century, Web 2.0 based OSNs are one of the impactful human innovations. OSNs have revolutionized the scope and experience of human communication by facilitating real-time information broadcasting. Around the world, nearly one in every two users is using OSN[1] and it is particularly high among the young generation. OSNs facilitate the users in information sharing, connectivity, and entertainment, but adversaries also misuse them for malicious activities. The large user-base, real-time message broadcasting, anonymity, and open nature have made OSNs as facilitating platforms for malicious activities like trolling, astroturfing, spamming, and rumor. Such malicious activities are generally performed using fake profiles in the form of bots, human-assisted cyborgs, Sybil, and compromised accounts. Now, OSN platforms are facing newer threats like astroturfing, propaganda, spear phishing, which are more sophisticated in comparison to the traditional cyber threats like spamming, DDoS attack, and identity theft. Among various malicious actors, automated profiles (*aka* socialbots) are responsible for various modern illicit activities, such as *political astroturfing* [1] and *rumor* [2]. Varol et al. [3] have estimated that approximately 9% to 15% of Twitter accounts are socialbots. Though socialbots can be used for both positive and negative activities, their disruptive and malicious role in major political events like in 2016 USA presidential election, Brexit referendum [1], Arab spring protest [4] has labeled them as devils. As a result, researchers are analyzing different malicious aspects of socialbots [5]. In addition, researcher have injected socialbots in OSNs to investigate their potential [6].

In the existing literature, most of the popular and efficient methods for socialbots detection are based on supervised learning, wherein a set of hand-crafted features from various categories are defined, and machine learning models are trained to segregate the bots and benign users [7]–[9]. The feature engineering process is a manual and time-consuming task incorporating human biases and deficiencies. In the second line of approaches, researchers have proposed graph partitioning-based methods to detect the groups of socialbots. Apart from *feature engineering* and *graph partitioning*-based approaches, researchers have proposed behavior and temporal modeling-based approaches for socialbots detection [10]–[12]. Although behavior modeling-based approaches are independent of feature engineering, they model users exploiting only one behavior, either temporal [11], [12] or posting type [10]. Therefore, these approaches can detect either one or other type of socialbots, not both. To circumvent the complicated process of feature engineering, researchers are using the advancement in deep neural networks towards socialbots detection. However, it is still unexplored except few approaches [13], [14]. These approaches have not utilized all the aspects of users to model their automation; e.g., [13] is based on only profile information, whereas [14] exploits only temporal and content information. This paper fills this vacuum and models OSN users integrating the cues from their *profile*, *temporal*, *activity*, and *content* information and model them at two levels of granularity. Further, it applies hierarchical attention mechanism at both the level of user representations to assign variable weight to each information depending on their importance. This paper has three levels of novelty – (i) it exploits the strength of feature engineering through a simple representation of different information using vectors of raw data like *profile information* and avoids the tedious task of human-assisted feature crafting, (ii) unlike traditional behavior modeling approaches of presenting models based on only one behavior, proposed model integrates the cues from four aspects of users behavior, and (iii) finally integrate the cues from *(i) and (ii)* and fuse them with a *hierarchical attention-based*

[1] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

neural network model exploiting the strength of modern deep neural network techniques to find the regularities in sequential data representing the users behavior.

## II. Related Work

The existing literature has several approaches for socialbots detection which can be classified into different categories. Based on the methodology, we have classified the socialbots detection methods into four categories, namely (i) feature engineering-, (ii) graph partitioning-, (iii) behavior modeling-, and (iv) deep learning-based methods. A brief review of literature of each of these categories of methods is presented in the following paragraphs.

In feature engineering-based approaches, hand-crafted features are designed based on information extracted from users' profile, connection network, textual content, and other activities, and machine learning models are trained on the labeled training dataset [15], [16]. Service providers have also started developing their adversarial detection approaches like Facebook developed Facebook Immune System [17], which monitors every read and write actions on the database and label them as malicious or benign. Authors in [9], [18] have used *profile-*, *content-* and *interaction*-based features to segregate the spambots from benign users and evaluated it over different OSN datasets. In one of the most efficient and popular approaches, Devis et al. [7] designed more than 1000 features to train a random forest classification model to label an OSN user as socialbot or benign. Unlike [19], [20], most of the existing machine learning approaches characterize users based on their attributes, instead of characterizing users based on their connecting users and associated communities. To this end, [8] proposed a machine learning approach for socialbots detection on Twitter. Feature engineering-based classification systems can be easily circumvented by socialbots through manipulating their behavior as per the defining features of the underlying detection system. Additionally, handcrafting of features is a time consuming and tedious task. The second category, graph partitioning-based approaches assume that Sybil or socialbots cannot create sufficient connections with benign users, and researchers exploit this intuition to model and partition the connection network/graph. The existing literature has several approaches during the early days of the problem [21], [22]. Cao et al. [22] presented a scalable and computationally efficient method, *SybilRank*, to define the likelihood of OSN users of being malicious based on their structural properties in a manner that malicious users are ranked lower. Cao and Yang [23] presented an efficient Sybil defense system employing negative feedback on users to detect attack edges. In [24], the authors first presented an analysis of a socialbot injection experiment. Further, they modeled the OSN users as a graph and applied the Markov clustering to detect the socialbots operating in a coordinated manner. The existing literature also perform the behavior modeling of user activities for detecting socialbots. Zhang et al. [25] performed Pearson $\chi^2$-test on users tweets-time for detecting automated accounts. Pan et al. [12] presented a *burstiness*-based approach to categorize the socialbots and benign users. Similarly, existing literature has a number of other behavior modeling-based approaches for socialbots detection [10], [11], [26].

Recently, researchers have started exploiting the advancement in the deep learning techniques for socialbots detection. In a first approach of this kind, [14] jointly modeled the *behavior* and *textual* information of users towards socialbots detection. Similarly, Ping et al. [27] presented another deep learning-based model for bots detection with a joint modeling of users based on their *temporal* and *content* information. These two approaches do not incorporate any information regarding the type of activities such as *tweet*, *retweet*, *quoted tweet* performed by the users. Further, they have neither exploited the profile information nor modeled the sequence of time intervals between consecutive tweets, which have already been used by many researchers for bots characterization and detection [11]–[13]. In another approach, [13] just used profile information along with SMOTE [28] to detect the socialbots. However, none of the deep learning-based approaches has utilized the hierarchical attention mechanism to assign variable weights at two levels of granularity.

## III. Proposed Model

Figure 1 presents the proposed hierarchical attention-based deep neural network model for socialbots detection which is described in the following sub-sections.

### A. Low-Level Feature Encoding

This layer extracts fine-grained users behavior and characteristics based on their *profile*, *temporal*, *activity*, and *content* information. The sequential pattern using *profile*, *temporal*, and *activity* information is modeled using BiLSTM while content information is encoded using CNN. In sequential modeling, LSTM only incorporates historical information from a sequence avoiding the future information. BiLSTM resolves this issue incorporating both the historical and future contextual information of a sequence. Further, deep RNN has better low-level feature representation and greater model complexity [29]. Therefore, to incorporate historical and future contextual information on sequential data along with having better low-level feature representation, this paper uses two-layer stacked BiLSTM. For an input feature vector $f$, the forward component of BiLSTM process it from $f_1$ to $f_n$ to get the historical context whereas backward component process it from $f_n$ to $f_1$ to get the future context. Mathematically, for each $f_i \in f$, the historical and future contexts-based hidden states and its annotation are learned as defined in Equations 1-3. Similarly, CNN process data in tabular form that consists of two layers – *convolution* and *pooling*. In *convolution* layer, convolution operation is performed to extract high-level *feature map* out of the input tabular data, whereas *pooling* layer performs pooling operation to extract important features from the *feature map*.

$$\overleftarrow{h}_i = \overrightarrow{LSTM}(f_i) \tag{1}$$

$$\overrightarrow{h}_i = \overleftarrow{LSTM}(f_i) \tag{2}$$

$$h_i = [\overleftarrow{h}_i, \overrightarrow{h}_i] \tag{3}$$

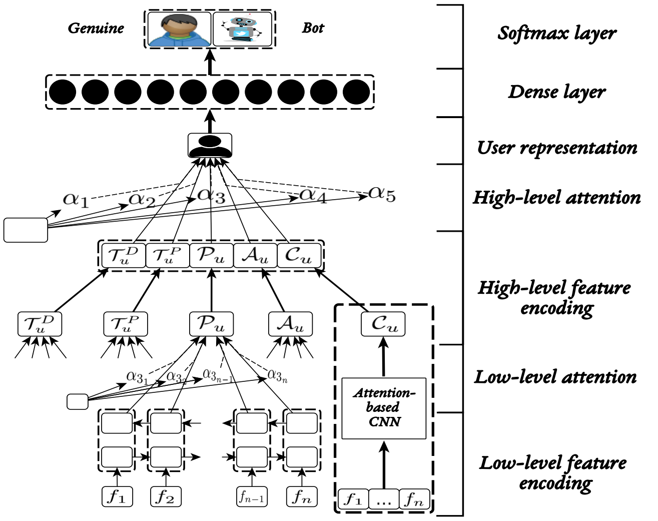

Fig. 1. Hierarchical attention network-based deep neural network model

*1) Profile Representation:* To evade the detection mechanisms, generally *botherder* manually create profiles to look them real. However, vital information such as *follower count*, *following count*, *status count* are based on their automation mechanism; therefore, these information will still exhibit certain pattern [13]. To detect the socialbots having regularities in their simple profile information, our proposed approach models the users identity-related information using 10 basic features like *deafult profile status*, *verified or not*, *follower count*. To avoid manual feature crafting, the proposed model uses information directly extracted from the user profiles. The profile feature vector of an OSN user $u$ is denoted by $P_u$.

*2) Temporal Behavior Representation:* To observe the regularity and automation of accounts, modeling of tweets time is vital. In terms of temporal behavior, bots are – (i) either programmed to be active at a specific point of time representing their *diurnal pattern* or (ii) they are programmed to be active after some specific time-interval representing their *periodic pattern*. The proposed approach models the two types of the temporal pattern of a user $u$ using temporal information of either the 30 day activities (if available) or a minimum of 100 activities. The above two types of modeling uses the temporal information of all the four types of tweeting activities – *plain tweet*, *retweet*, *quoted tweet*, *reply*. The modeling of two types of the temporal behavior of $u$ is presented in the following paragraphs.

(a) *Diurnal Behavior Representation:* We divide the `Twitter` timeline of $u$ in days, wherein each day $d \in D$ is divided into a sequence of time-intervals of 1 minute, say, $\tau = \{\tau_1, \tau_2, \ldots, \tau_{1440}\}$ because 1 day=24 × 60 minutes, where $D$ is the number of days between the first and last crawled tweets of $u$ used in this study. Thereafter, number of *tweeting activity* by $u$ in a particular minute of the day $d$ is assigned to the respective minute of the day vector $d_u$. For example, if $u$ performs 2 retweeting activity, posts 1 tweet, and replies to a tweet during 1 minute time-interval between $1:30$ to $1:31$ AM i.e. $91^{th}$ minute of $i^{th}$ day $d^i$, then $91^{th}$ index of $i^{th}$ day temporal vector $d_u^i$ is assigned a vlaue of 4. Finally, diurnal temporal behavior representation of $d_u^i$ is $\{dv_1^i, dv_2^i, \ldots, dv_{1440}^i\}$, where $dv_1^i$ represents the number of activities performed by $u$ in the first minute of $d^i$. Similarly, *diurnal temporal vector* is generated for each of the 30 days if the tweets are available otherwise zero vectors are created. Finally, we concatenate the *diurnal temporal vector* of each day of $u$ to create the final *diurnal temporal vector* $D_u$ of $u$ as given in Equation 4.

$$D_u = \{d_u^1, d_u^2, \ldots, d_u^{30}\} \tag{4}$$

(b) *Periodic Behavior Representation:* As discussed, bots can be programmed to be active at a fixed interval of time representing the periodic pattern in the inter-tweets time. This behavior of users is modeled to segregate the socialbots and benign users. To this end, we compute the *inter-tweets* time between every pair of consecutive activities. Given a user $u$, let $T_1$ and $T_2$ represents the times of tweets activities $t_1$ and $t_2$, respectively, then $\Delta T_{(1,2)} = T_2 - T_1$ represents the *inter-tweets* time between tweets $t_1$ and $t_2$. Similarly, we find the *inter-tweets* time between every pair of consecutive tweets and the resultant set of all $\Delta T$ over a set of $N$ tweets of $u$ is represented using $T_u$, as given in Equation 5.

$$\Delta T_u = \{\Delta T_{(1,2)}, \ldots, \Delta T_{i,i+1}, \ldots, \Delta T_{N-1,N}\} \tag{5}$$

*3) Activity Behavior Representation:* In OSNs, botherders program socialbots to perform different activities like tweets, retweets, replies, quoted tweets, or a combination of these, as per the requirement. Though socialbots can perform every possible activity like benign users, they will follow certain distribution regarding when and what type of activity to perform next? We selected a maximum of 100 tweets for each user to encode their activity behavior. An equal number of activities are selected for each user to ensure that the length of the encoded activity behavior representation for each user is the same. Each of the four activities is encoded by a unique number – *plain tweet* by 0, *retweet* by 1, and *reply* by 2, *quoted tweets* by 3. For example, given the recent 100 activities from the timeline of a user $u$, its activity-based behavior can be encoded as $\{0_1, 1_2, 1_3, 0_4, \ldots, 2_{100}\}$, where $1_3$ represents that $3^{rd}$ activity of $u$ is a *retweet*. Formally, the activity behavior representation of $u$ is shown in Equation 6, where $a_i$ represents the encoded value corresponding to the $i^{th}$ activity.

$$A_u = \{a_1, a_2, \ldots, a_i, \ldots, a_{100}\} \tag{6}$$

*4) Content Behavior Representation:* In existing literature, content quality has always been used as an indicator of malicious behavior [8]. Given, a user $u$ has posted $N$ tweets, say, $t_1, t_2, \ldots, t_n$, first, tweets are sorted in chronological order. Thereafter, each tweet $t_i$ is tokenized using keyspace and each token (word) $w \in W$ is replaced with $d$-dimensional `GloVe` embedding vectors. As a result, $i^{th}$ tweet $t_i$ of $u$ is represented as a matrix $m_{ui} \in \mathbb{R}^{|W| \times d}$. Similarly, each of the $N$ tweets of $u$ is represented as 2-dimensional word-embedding matrix. Finally, all the matrices are concatenated and represented as a 3-dimensional matrix $M_u \in \mathbb{R}^{N \times |W| \times d}$, which is passed through a six-layer CNN containing 2 max pooling layers, placed after $2^{nd}$ and $4^{th}$ convolutional layers, respectively. Each convolution layer contains $k$ filters, each of size $s \times s$. The tweets are organized as a 3D matrix so that it can learn both the *intra-* and *inter-tweets* pattern. The encoded content behavior representation of $u$ is denoted by $C_{ou}$.

### B. Low-Level Attention

Not each value of a particular behavior vector contributes equally to that behavior. To this end, we use the attention mechanism to assign variable weights to different values of a particular behavior vector. To this end, first, each behavior vector except $C_{ou}$ is passed through a two-layer stacked BiLSTM to learn the annotation of its features using Equations 1-3. Thereafter, the annotation of a feature is passed through a dense layer to get its hidden state representation using Equation 7, where $w$ and $b$ represent the weight and bias, respectively.

For example, profile vector $P_u$ is passed through a two-layer stacked BiLSTM to get the encoded representation of its features using Equations 1-3. Suppose, the hidden state representation of a particular feature $f \in P_u$ is $h_f$, then it is given to a dense-layer to learn the hidden representation, $h'_f$ using Equation 7. Thereafter, similarity between $h'_f$ and a vertex vector $v_f$ is calculated to find the importance of $f \in P_u$ using Equation 8, where $v_f$ is the high-level representation of the feature's importance which is randomly initialized and jointly learned during the training process [30]. Finally, attention-based weighted representation of $P_u$ is learned as a weighted sum of the hidden representation of each $f \in P_u$ using Equation 9 and it is represented as $\mathcal{P}_u$. Similarly, the same procedure is repeated for $D_u$, $\Delta T_u$, $A_u$ to learn the attention-based weighted representation $\mathcal{T}_u^D$, $\mathcal{T}_u^P$, and $\mathcal{A}_u$ respectively. Furthermore, in case of content behavior representation, $C_{ou}$ is passed through a six-layer CNN followed by attention mechanism to learn the attention-based weighted representation $C_u$.

$$h'_f = \tanh(wh_f + b) \tag{7}$$

$$\alpha_f = \frac{exp(h'_f v_f)}{\sum_f exp(h'_f v_f)} \tag{8}$$

$$\mathcal{P}_u = \sum_f (\alpha_f h_f) \tag{9}$$

### C. High-Level Feature Encoding

Following the modeling of five behavioral aspects of a user $u$, this section concatenates the low-level feature representation from section III-B together such that $B_u = \{\mathcal{P}_u, \mathcal{T}_u^D, \mathcal{T}_u^P, \mathcal{A}_u, \mathcal{C}_u\}$.

### D. High-Level Attention

Each of the five behavior components does not have equal contribution towards socialbots detection. Therefore, to assign a weight to each behavior component $B_u^c \in B_u$, we apply high-level attention using Equations 10 and 11, where $v_c$ is the high-level context vector, randomly initialized and jointly learned during the training process. The final representation of $u$, represented as $\mathcal{U}_u$, is the weighted sum of each of the five components as given in Equation 12.

$$B_u^{c\prime} = \tanh(wB_u^c + b) \tag{10}$$

$$\alpha_c = \frac{exp(B_u^{c\prime T} v_c)}{\sum_\beta exp(B_u^{c\prime T} v_c)} \tag{11}$$

$$\mathcal{U}_u = \sum_c \alpha_c B_u^c \tag{12}$$

### E. Bot Classification Layer

The final user vector $\mathcal{U}_u$ is the higher-level representation of a user comprising of all the features and passed through a softmax layer for classification.

## IV. Experimental Setup and Results

This section presents a brief description of evaluation datasets, applied experimental settings, evaluation results, and comparative analysis.

### A. Dataset Preprocessing

The proposed model is evaluated over the dataset provided by [31] which includes three socialbot datasets, namely, *social spambots#1* (*SD*$_1$), *social spambots#2* (*SD*$_2$), *social spambots#3* (*SD*$_3$), and one dataset of *genuine users* (*GD*). We evaluate the presented model over the three socialbot datasets individually to investigate its efficiency over different categories of socialbots. Moreover, we evaluate the robustness of the proposed model against different categories of socialbots over a fourth dataset, *SD*$_4$, by mixing the first three socialbot datasets. The reason behind the generation of *SD*$_4$ is that existing approaches detect well a particular category of socialbot but fail for other categories [32]. The evaluation datasets contain the users who have posted at least 100 tweets. In each of the four discussed datasets, genuine users (negative class) are selected from genuine dataset *GD*. Table I presents the statistics of datasets, wherein *original* column denotes the number of users in the original version of the underlying dataset and *final* column denotes the number of users extracted for evaluation after filtering the users having less than 100 tweets. We can analyze from the table that except *SD*$_2$, the ratio of socialbots and benign users is skewed. Therefore, we evaluate the proposed model on two versions of the datasets – balanced and original (imbalanced). In the balanced version, the number of benign users in each dataset is equal to the number of socialbots in the underlying dataset. In the case of *SD*$_1$ and *SD*$_3$, as the number of benign users is more than

TABLE I
STATISTICS OF THE DATASETS

| Dataset | #User accounts | |
|---|---|---|
| | Original | Final |
| $SD_1$ | 991 | 991 |
| $SD_2$ | 3457 | 2107 |
| $SD_3$ | 464 | 461 |
| $SD_4$ | 4912 | 3559 |
| *GD* | 3474 | 1973 |

socialbots, we randomly sample the equal number of genuine users from *GD* like random sampling of 991 genuine users for $SD_1$. However, in the case of $SD_2$ and $SD_4$, as the number of genuine users is less than socialbots, we generate the remaining number of genuine users using an oversampling method called SMOTE. In the case of the imbalance evaluation, we take the available number of socialbots and genuine users. For example, in the case of $SD_1$, the number of socialbots and genuine users will be 991 and 1973, respectively.

TABLE II
PERFORMANCE EVALUATION RESULTS OVER BALANCED AND IMBALANCED VARIANTS OF THE DATASETS

| Dataset | Results on balanced dataset | | | | Results on imbalanced dataset | | | |
|---|---|---|---|---|---|---|---|---|
| | DR | Pr | F1 | Acc | DR | Pr | F1 | Acc |
| $SD_1$ | 100.00 | 100.00 | 100.00 | 99.62 | 99.64 | 100.00 | 99.81 | 99.83 |
| $SD_2$ | 98.83 | 99.33 | 99.08 | 98.87 | 99.43 | 99.45 | 99.42 | 99.51 |
| $SD_3$ | 100.00 | 100.00 | 100.00 | 100.00 | 98.40 | 99.45 | 98.86 | 99.38 |
| $SD_4$ | 98.31 | 98.11 | 98.23 | 98.57 | 98.10 | 98.95 | 98.44 | 97.90 |

### *B. Training Details*

The presented approach uses 80% data to train the models, which are validated on the remaining 20% of the data. We used two-layer stacked BiLSTM, having 256 memory cells at each layer, as the user representation layer to learn *profile*, *temporal* and *activity* representations. We also used six-layer CNN having 256 filters of $3 \times 3$ at the first three layers, followed by a max pooling operation of $3 \times 3$. At the last three layers, 256 filters of $2 \times 2$ are further used, followed by a max pooling operation of $2 \times 2$. In CNN, we performed two-dimensional operation because a user's tweet content is mapped to a three-dimensional matrix like image representation. The three-dimensional content representation captures better latent features from *intra-tweet* and *inter-tweets* perspectives. The *user representation* layer follows a dense layer with 1024 neurons. The dense layer performs a dropout operation of 0.5 to reduce the over-fitting effect. The final layer is a softmax function having two neurons for classification. We use *adam* as an optimizer.

### *C. Performance Evaluation Results*

We evaluate the proposed model on the balance and imbalance versions of four socialbot datasets. Table II presents the performance evaluation results on the balance and imbalance variants of four datasets. We can observe that in the case of the balanced dataset, the proposed model consistently shows good results in terms of all the four evaluation metrics. We can also observe that even in some cases, the proposed model shows perfect performance. However, over the imbalanced dataset, the performance is slightly lower in comparison to the balanced dataset.

We also performed a comparative performance evaluation to establish the efficacy of the proposed model in classifying the socialbots and genuine users on `Twitter`. We performed comparative analysis with three baselines and three SOTA deep learning methods [13], [14], [27]. We implemented the comparison approaches employing all the parameters discussed in the respective papers. We also built three baselines using the proposed model based on different variants of neural networks. `Baseline-1` uses LSTM instead of BiLSTM without attention mechanism. `Baseline-2` uses LSTM with the attention mechanism. Finally, `Baseline-3` excludes the attention mechanism to observe its impact over the performance evaluation. We performed all the comparative performance evaluation over the original (imbalanced) version of the dataset. Table III presents the comparative performance evaluation results. We can observe that except for two, the proposed hierarchical attention-based model shows better performance in comparison to the comparative models for all cases. On investigation, we found that the `DNNBD` model [13] shows a highly inconsistent performance when we repeat the same experiment many times. Therefore, the average evaluation results of `DNNBD` are significantly low in comparison to the proposed model. In Table III, the best performance evaluation result for each case is shown in bold typeface. We can observe from the table that among the three comparative deep learning models, `DBDM` [14] shows the best and `DNNBD` [13] shows the worst performance. We can also observe that the baseline methods also show comparative performance because these are nearly similar to the presented model in terms of configuration. Furthermore, we can observe that the more similar a baseline is to the proposed model in terms of configuration like *Baseline-2*, the closer is its result to the proposed model.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a hierarchical attention-based neural network model for socialbots detection in OSNs. The proposed approach models an OSN user exploiting five different behavior representations extracted from *temporal*, *activity*, and *content* information. The user behavior is jointly modeled using CNN and BiLSTM architectures at two-levels of granularity. The *temporal* and *activity* information is modeled as sequences, which are given to a two-layer stacked BiLSTM, whereas *content* information is passed to a six-layer CNN. Additionally, hierarchical attention mechanism is applied at both low-level and high-level user representations. On experimental evaluation, the proposed model performs better in comparison to the five state-of-the-art socialbots detection methods of different categories. In the future, we plan to evaluate the proposed model on multiple datasets from different sources. We also plan to develop an API of the

TABLE III
COMPARATIVE PERFORMANCE EVALUATION RESULTS OF OUR PROPOSED MODEL

| Approach | $SD_1$ | | | | $SD_2$ | | | | $SD_3$ | | | | $SD_4$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DR | Pr | F1 | Acc | DR | Pr | F1 | Acc | DR | Pr | F1 | Acc | DR | Pr | F1 | Acc |
| Proposed model | **99.64** | **100.00** | **99.81** | **99.83** | **99.43** | **99.45** | **99.42** | **99.51** | **98.40** | **99.45** | **98.86** | **99.38** | **98.10** | 98.95 | **98.44** | 97.90 |
| DNNBD [13] | 73.72 | 77.66 | 75.63 | 78.20 | 79.50 | 79.62 | 79.56 | 79.72 | 43.38 | 80.32 | 56.33 | 66.48 | 64.21 | 69.82 | 66.89 | 66.68 |
| DBDM [14] | 97.83 | 100.00 | 98.82 | 99.32 | 98.66 | 98.00 | 98.25 | 98.80 | 94.27 | 98.26 | 95.89 | 98.54 | 95.71 | **99.05** | 97.29 | 95.85 |
| DeBD [27] | 97.49 | 97.73 | 97.59 | 97.74 | 98.39 | 99.10 | 98.76 | 99.12 | 95.54 | 96.01 | 95.81 | 97.92 | 95.40 | 96.21 | 96.66 | 95.26 |
| BotOrNot [7] | 86.10 | 83.27 | 84.66 | 96.66 | 98.82 | 98.46 | 98.63 | 98.44 | 88.71 | 86.88 | 87.78 | 96.69 | 96.92 | 98.69 | 97.79 | 97.30 |
| DeBot [11] | 2.03 | 42.85 | 3.87 | 89.22 | 0.12 | 33.33 | 0.23 | 42.50 | 2.62 | 55.55 | 5.00 | 86.66 | 0.50 | 71.42 | 0.99 | 38.30 |
| Baseline-1 | 98.89 | 97.91 | 98.93 | 98.37 | 97.80 | 99.12 | 98.81 | 98.89 | 95.00 | 96.42 | 95.53 | 98.76 | 96.16 | 97.94 | 97.11 | 96.12 |
| Baseline-2 | 99.19 | 98.39 | 98.79 | 98.78 | 98.46 | 99.30 | 98.87 | 99.01 | 96.41 | 99.26 | 97.61 | 98.76 | 95.06 | 98.95 | 96.88 | 95.79 |
| Baseline-3 | 98.38 | 96.91 | 97.64 | 97.61 | 98.10 | 99.06 | 98.65 | 98.83 | 98.34 | 96.83 | 97.92 | 98.97 | 97.01 | 98.76 | 96.74 | **99.55** |

proposed model that can be used by the academic community and others for research purposes.

## REFERENCES

[1] A. Bessi and E. Ferrara, "Social bots distort the 2016 u.s. presidential election online discussion," *First Monday*, vol. 21, no. 11, 2016.

[2] C. Shao, G. L. Ciampaglia, O. Varol, K. C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature Communications*, vol. 9, no. 11, 2018.

[3] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization." Montreal, Canada: AAAI, 2017, pp. 280–289.

[4] N. Abokhodair, daisy Yoo, and D. W. McDonald, "Dissecting a social botnet: Growth, content and influence in twitter," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*. Vancouver, BC, Canada: ACM, 2015, pp. 839–851.

[5] M. Abulaish and M. Fazil, "Socialbots: Impacts, threat-dimensions, and defense challenges," *IEEE Technology and Society Magazine*, vol. 39, no. 3, pp. 52–61, 2020.

[6] M. Fazil and M. Abulaish, "Why a socialbot is effective in twitter? a statistical insight," in *Proceedings of the 9th International Conference on Communication Systems and Networks (COMSNETS), Social Networking Workshop*. Bengaluru, India: IEEE Computer Society, 2017, pp. 562–567.

[7] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in *Proceedings of WWW*. Montreal, Canada: ACM, 2016, pp. 273–274.

[8] M. Fazil and M. Abulaish, "A hybrid approach for detecting automated spammers in twitter," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2707–2719, 2018.

[9] F. Ahmed and M. Abulaish, "A generic statistical approach for spam detection in online social networks," *Computer Communications*, vol. 36, no. 10, pp. 1120–1129, 2013.

[10] S. Cresci, R. D. Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Dna-inspired online behavioral modeling and its application to spambot detection," *IEEE Intelligent System*, vol. 31, no. 5, pp. 58–64, 2016.

[11] N. Chavoshi, H. Hamooni, and A. Mueen, "Debot: Twitter bot detection via warped correlation," in *Proceedings of ICDM*. Barcelona, Spain: IEEE Computer Society, 2016, pp. 817–822.

[12] J. Pan, Y. Liu, and H. Xiang Liu, "Discriminating bot accounts based solely on temporal features of microblog behavior," *Physica A: Statistical Mechanics and its Applications*, vol. 450, no. 3, pp. 193–204, 2016.

[13] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Information Sciences*, vol. 467, no. 10, pp. 312–322, 2018.

[14] C. Cai, L. Li, and D. Zeng, "Detecting social bots by jointly modeling deep behavior and content information," in *Proceedings of CIKM*. Singapore: ACM, 2017, pp. 1995–1998.

[15] A. H. Wang, "Detecting spam bots in online social networking sites: A machine learning approach," in *Proceedings of the 24th International Conference on Data and Applications Security and Privacy*. Rome, Italy: Springer, 2010, pp. 335–342.

[16] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the ICCSA*. Austin, Texas, USA: ACM, 2010, pp. 1–9.

[17] T. Stein, E. Chen, and K. Mangla, "Facebook immune system," in *Proceedings of 4th International Workshop on Social Network Systems*. Salzburg, Austria: ACM, 2011, pp. 1–8.

[18] C. Yang, R. C. Harkreader, and G. Gu, "Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers," in *Proceedings of RAID*. Menlo Park, California, USA: Springer Berlin Heidelberg, 2011, pp. 318–337.

[19] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving twitter spammers," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1280–1293, 2013.

[20] S. Y. Bhat and M. Abulaish, "Community-based features for identifying spammers in online social networks," in *Proceedings of ASONAM*. Ontario, Canada: ACM, 2013, pp. 100–107.

[21] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *Proceedings of 16th International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New Delhi, India: ACM, 2010, pp. 363–374.

[22] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proceedings of 9th International Conference on Networked Systems Design and Implementation*. Chicago, USA: USENIX Association, 2012, pp. 1–14.

[23] Q. Cao and X. Yang, "Sybilfence: Improving social-graph-based sybil defenses with user negative feedback," Dept of Computer Science, Duke University, Tech. Rep., 2012.

[24] M. Fazil and M. Abulaish, "A socialbots analysis-driven graph-based approach for identifying coordinated campaigns in twitter," *Journal of Intelligent & Fuzzy Systems,*, vol. 38, no. 3, pp. 2961–2977, 2020.

[25] C. M. Zhang and V. Paxson, "Detecting and analyzing automated activity on twitter," in *Proceedings of 7th International Conference on Passive and Active Network Measurement*. Atlanta, USA: Springer, Berlin, Heidelberg, 2011, pp. 102–111.

[26] Z. Chen, R. S. Tanash, R. Stoll, and D. Subramanian, "Hunting malicious bots on twitter: An unsupervised approach," in *Proceedings of 8th International Conference on Social Informatics*. Oxford, United Kingdom: Springer, Cham, 2016, pp. 501–510.

[27] H. Ping and S. Qin, "A social bots detection model based on deep learning algorithm," in *Proceedings of ICCT*. Chongqing, China: IEEE Computer Society, 2018, pp. 1435–1439.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[29] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proceedings of ICLR*. Banff, Canada: Arxiv, 2014, pp. 1–13.

[30] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT*. California, USA: ACL, 2016, pp. 1480–1489.

[31] S. Cresci, R. D. 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 WWW*. Perth, Australia: ACM, 2017, pp. 963–972.

[32] J. Echeverria, E. D. Cristofaro, N. Kourtellis, I. Leontiadis, G. Stringhini, and S. Zhou, "Lobo-evaluation of generalization deficienciesin twitter bot classifiers," in *Proceedings of ACSAC*. San Juan, USA: ACM, 2018, pp. 137–146.