A Link Analysis Based Approach to Predict Character Death in Game of Thrones

Swati Agarwal[1(✉)], Rahul Thakur[2], and Sudeepta Mishra[3]

[1] BITS Pilani, Goa Campus, Goa, India
agrswati@ieee.org
[2] IIT Roorkee, Roorkee, India
rahul@ieee.org
[3] BITS Pilani, Hyderabad Campus, Hyderabad, India
sudeepta@hyderabad.bits-pilani.ac.in

Abstract. Mysterious and uncertain deaths in the "Game of Thrones" novel-series have been stupefying to the vast pool of readers and hence interested researchers to come up with various models to predict the deaths. In this paper, we propose a Death-Prone Score model to predict if the candidate character is going to die or stay alive in the upcoming book in the series. We address the challenge of high-dimensional data and train our model on the most significant attributes by computing feature importance in the vector space. Further, we address the challenge of multiple interactions between characters and create a social network representing the weighted similarity between each character pair in the book. The proposed model takes similarity and proximity in a social network into account and generates a death-prone score for each character. To evaluate our model, we divide the characters data into training (characters died before year 300) and testing (characters died in the year 300 and characters alive till year 300). Our results show that the proposed Death-Prone Score model achieves an f-score of 86.2%.

Keywords: Character similarity · Death prediction · Feature importance · Game of Thrones · Social network analysis · Weighted vector space model

1 Introduction

'A Game of Thrones' is a novel series were written by renowned American novelist George R. R. Martin[1] can be regarded as the most popular book series of the contemporary generation if not of all time. The novel deals with various royal families, diplomats, and bureaucrats with a blending of numerous major

[1] http://www.georgerrmartin.com/grrm_book/a-game-of-thrones-a-song-of-ice-and-fire-book-one/.

H. Gao et al. (Eds.): TridentCom 2019, LNICST 309, pp. 229–244, 2020.
https://doi.org/10.1007/978-3-030-43215-7_16

and minor characters instead of a handful of characters which adds to its dimensions [11]. Among various occurrence trends in the series, sudden, abrupt, and mind-boggling deaths have been fascinating events for the readers. For instance, in the first book of the series, *Ned Stark* appeared to be the main protagonist of the series. However early, in the beginning, he is beheaded by a boy *King Joffrey* signalling that anyone can die in the series at any time. Such sudden deaths led to great impacts in the series as well as among the audiences [18]. In this particular case, the beheading of *Ned Stark* led to the origin of *The War of Five Kings*[2] while there was a buzz on social media as well as on the relevant blogs and forums[3]. Following the sequence of many such sudden incidents (death of *Khal Drogo*, *King Joffrey*, and *King Robb Stark*), audiences started guessing and gauging the upcoming deaths, trying to understand author's writing patterns, character psychology, and interrelations. The immense popularity of the series and many such unexpected events stirred researchers and analysts to analyze this interesting series.

Background and Motivation: We conduct a literature survey in three lines of research: (1) studies conducted on Game of Thrones (GoT) books or tv series, (2) predicting characters' death for GoT as well as other similar series, and (3) character analysis conducted on other movies, tv-series, and books. Beveridge et al. [2] performs a context-based analysis on the books and create a social network of all characters and their houses. They use social network measures and compare their results against the importance of each character in the respective book. Inspired by the above article, Bonato et al. [4] performed a social network analysis on four different books; Twilight by S. Meye, S. King's The Stand, and J.K. Rowling's Harry Potter and the Goblet of Fire. They used social network measures and supervised learning-based models to investigate character modelling in these books and identify the most influential characters. Over the last two years, many researchers conducted experiments on the dataset of GoT series and published their results on blogs. Janosov [10] created a social network (approximately 400 nodes and more than 3000 edges) of all the characters pairs based on their interaction (appearing together in a scene). He used various network measures (PageRank, degree, weighted degree, clustering, betweenness, closeness, and eigen centralities) as features and applied SVM algorithm for predicting the next character to be dead in the series. Similarly, Phillip Tracy used the application of modularity and clustering network measures to predict the death risk of a character before an event happened [20]. In [19], Joel Shurkin surveys various social network analysis based approaches proposed for 18^{th} century English novels and concludes that the degree distribution of a character in the network follows the Power Law [9]. Unlike other book series such as *Lord of the Rings*, *Twilight*, or *Harry Potter*, predicting characters' death in *Game of Thrones* is a technically challenging due to the following three major reasons: (1) the vast domain of characters, complex relation, and ratio of their importance with their interactions, (2) automating the interpretation of the interactions of

[2] https://awoiaf.westeros.org/index.php/War_of_the_Five_Kings.
[3] https://www.vox.com/2016/6/1/11669730/tv-deaths-character-best.

characters (based on their appearance in a scene, similar houses, etc.), and (3) the absence of ground truth for upcoming portions of the series. Furthermore, the existing studies ignore the fact of the relative importance of different types of interactions between characters. For example, the importance of belonging to the same "house" over fighting as an opponent in the same "battle". Inspired by the existing literature, we formulate the problem of character death prediction as network analysis and classification problem. Social Network Analysis on the network of characters of Game of Thrones depicts different natural and behavioural aspects of the actors, thus providing a logical base of analysis. Understanding the relationship and network bindings between the dead characters and the remaining ones provide us with a likelihood of an existing actor's death. The Algorithmic approach lays a mathematical foundation of depicting proneness of a character's death in the upcoming series.

2 Novel Research Contributions

In contrast to the existing work, the work presented in this paper makes the following novel contributions:

1. ***A utility matrix-based model for generating a character profile:*** While existing studies use one relation as a notion of similarity between characters, we propose to create a character profile from all attributes present in the character-attribute utility matrix. We further assign the weights to each attribute based on their importance in predicting the death of a character in the upcoming novel.
2. ***A link analysis-based model for computing the similarity between character profiles:*** We create a social network of all the characters and propose a Death-Prone Score (DPS) model that assigns a score to each node (or character) in the network. The proposed model is a function of proximity (distance in the graph) and similarity (character profile) with connecting dead nodes for predicting the death of the candidate character.
3. ***A probabilistic model to compute DPS for disconnected profiles:*** We design a joint probability and damping factor based DPS model to address the challenge of no direct edge between a dead and alive character pair. Such model removes the biases in the prediction that only the nodes within proximity to the dead node are highly likely to die next unlike the existing studies where centrality is the primary measure used for the prediction.

Computing DPS is not limited to the predicting the death of a character in a novel or television series, but it has its applications in many domains [15]. Therefore we expect our results to be useful for the communities that analyze the impact of one event on another linked in a network (similar to knock-on effect in percolation theory) [14,17]. For example, we expect our model to be useful for cellular network users or operators to predict the failure of a device based on the failure of some existing devices in the network. The proneness of failure can be predicted by computing the weighted similarity between the failed and active devices [22]. Similarly, in case of a malicious attack happened in a network, the proposed approach can be used to detect the devices to be attacked next. Also, based on the similarity between user profiles hacked on a social network, the model will be able to capture the next accounts to be hacked. Similarly, based on the information about some channels posting fake news on social media and their links with other channels can be exploited to determine the diffusion of fake news and the identification of potential channels uploading fake news [21].

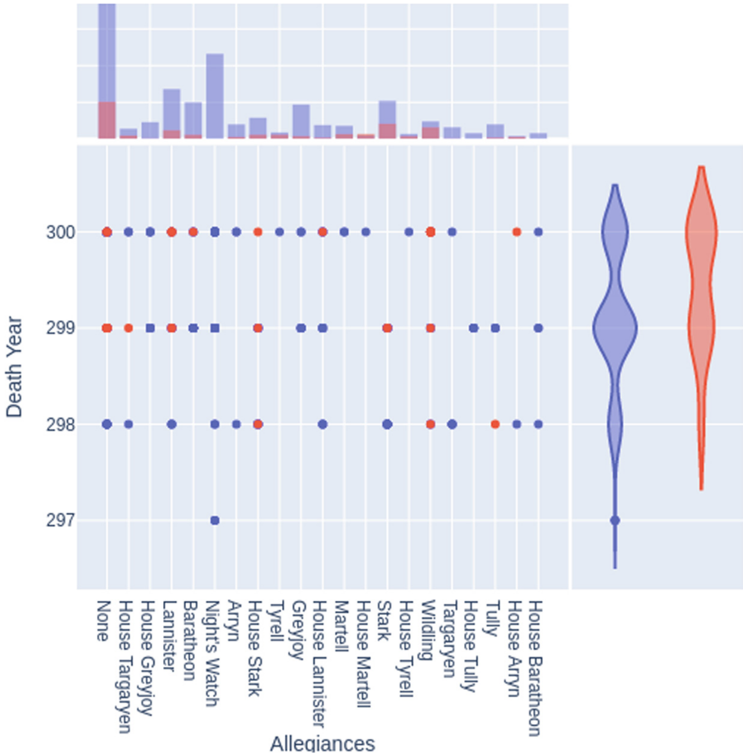

Fig. 1. Illustrating the statistics of the dataset in character-deaths.csv file. X-axis represents the various allegiances in the book and their death year is represented on Y-axis. The Violin plot on the right side and the histogram plot on the top represent the individual statistics of death year and allegiances respectively.

3 Experimental Setup

We conduct our experiments on an open source data downloaded from Kaggle[4]. The available dataset is created by combining multiple datasets initially collected by different groups and teams and published online for public use. The dataset on Kaggle consists of three CSV files (battles.csv, character-deaths.csv, and character-predictions.csv) collectively created for first four books

[4] https://www.kaggle.com/mylesoneill/game-of-thrones.

from the series (A Game of Thrones, A Clash of Kings, A Storm of Swords, and A Feast for Crows).

- **battles.csv:** This file is a comprehensive collection of all the battles fought in the series in four books. The dataset is created from the "*The War of the Five Kings*" dataset[5] collected initially by Chris Albon. The dataset contains details about 38 battles fought in three years in 10 different regions. The dataset includes a total of 25 attributes including battle name, attacker king, defender king, region, location, and year of the battle, number of major deaths, the outcome of the battle, etc. The dataset reveals that there are four different types of battles while the number of participants information is available only for three types of battles. Similarly, for some of the battles no information is available on the attacker or defender kings causing missing values in our dataset. Since the dataset is generated from the original book scripts, the missing values cannot be filled using pre-processing or statistical measures.
- **character-deaths.csv:** This file contains the exhaustive list of the characters and when they died. The file was created by Erin Pierce and Ben Kahle as a part of their Bayesian Survival Analysis [16]. The file contains a total of 917 characters featured using 13 attributes such as year of death, book and chapter number, appearances in different books in the series, etc. We conduct an exploratory data analysis to identify some features that are useful for the prediction model. Figure 1 shows the basic statistics of the exploratory data analysis. The scatter plot in the Figure shows characters died in Year 297–300 from various house groups (None represents the unknown house information). The Violin graph on the Y-margin represents the number of characters died in each year filtered based on the gender of the characters. The violin graph reveals that maximum number of male characters died in year 299. While, the maximum number of female characters died in year 300 while in 297 year, no female characters had died. We combine this meta information with the histogram plotted in X-margin. The overall graph reveals that the majority of the characters who died in first three books do not have their house (Allegiances) information available in the dataset irrespective of their gender. The graph also reveals that maximum number of male characters who died in year 297–300 are from Night's Watch house followed by Lannister and Baratheon groups with relatively higher ratio of female characters. The variation in number of characters died from a house and the ratio of female and males indicates that allegiances and the gender are the two strong indicators for predicting the next characters to die in the series.
- **character-predictions.csv:** This file contains the comprehensive list of all 1946 characters and their metadata. For example, *name*, *house*, *culture*, *title*, *family relations* (parents, siblings, spouse), *dead* or *alive*, *title*, *gender*, date of birth, etc. This dataset was created by the team '*A Song of Ice and Data*' and later made available for research community on Kaggle[6].

[5] https://github.com/chrisalbon/war_of_the_five_kings_dataset.
[6] http://awoiaf.westeros.org/index.php/Main_Page.

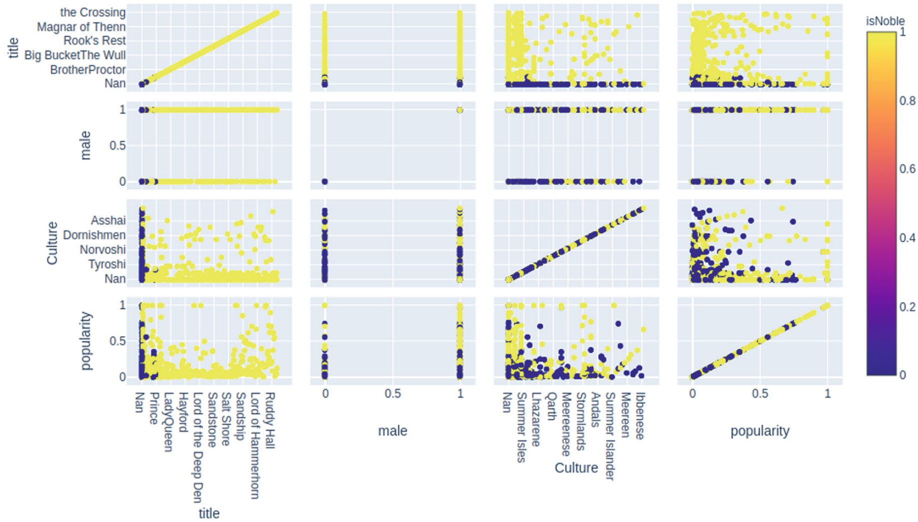

Fig. 2. A matrix based graph amongst various features from the dataset. The relation between attributes is exploited to determine the weight and priority of each feature for the prediction mode. The color parameter in the graphs shows the binary score of attribute IsNoble representing whether the character who died with different characteristics (popularity, culture, title, and gender) belong to noble class or not. (Color figure online)

We merge all the files to create our experimental dataset. The dataset consists of 53 attributes including four categorical and 49 numeric attributes. We convert each categorical attribute into a numeric format. Since *name* is a nominal categorical attribute, we assign a unique ID to each name. For *title*, *house*, and *culture* attributes, we convert them to an ordinal attribute and assign a priority to each field based on their frequency within the attribute. If a field (title type, house type, or culture type) is more frequent in the dataset (across all characters), we decrease its priority over the fields occurring less frequently. Figure 2 shows the graph of relation between various characteristics from utility matrix. The graphs are further labeled based on the status of the characters (isNoble = 0 or 1). The sub-graph [3, 1] demonstrates the relationship between various titles and cultures. The colors in the sub-graph [3, 1] reveals that any dead character that has a title or a culture is possibly a noble as well. Whereas, all the characters with unknown title or unknown culture are highly likely to be a no-noble person. The sub-graphs [4, 3] and [3, 4] reveals that many characters who have an unknown culture are still popular in the book while majority of them are male characters. Whereas, there are many female characters who belong to a culture and have high popularity as well. The sub-graph [3, 4] reveals that there are many no-noble characters who are more popular in the series in comparison to the characters who belong to a particular culture. The sub-plot [2, 3] shows that across all the cultures, majority of the dead female characters are not from noble house. Similarly, sub-plot [2, 4] reveals that most popular characters

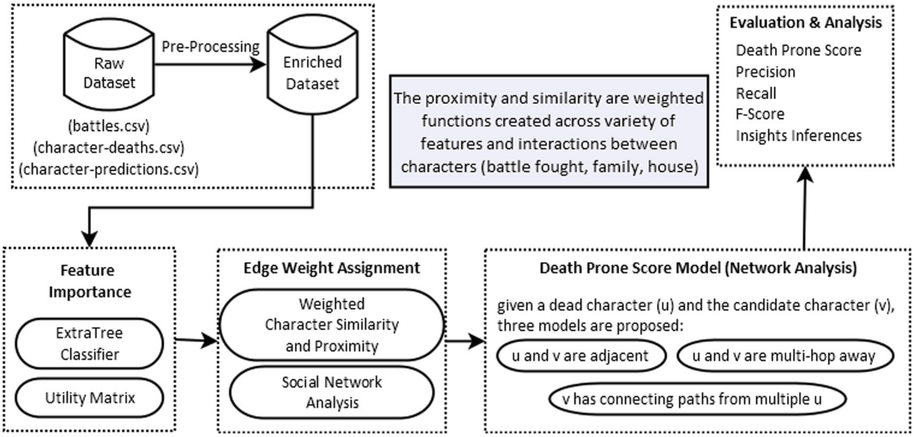

Fig. 3. A general high-level framework proposed for the prediction of dead and alive characters in the dataset.

who belonged to noble family are male while only a very few female characters are from noble family and popular as well. Otherwise, majority of the female characters who died are neither popular nor belong to the noble family.

We further normalize all priority scores to avoid the dominance in high-value attributes. We use the following function to compute the priority of each attribute value within an attribute: $Priority_x = f(x) = \frac{1}{TF_x} \times k$, where TF_x represents the column-wise frequency of x attribute value, and k represents the normalizing constant. The final dataset after merging consists of 1947 characters. To conduct our experiments, we divide our dataset into training and testing data. Since the year 300 is the most recent year in the downloaded dataset we use the characters present in the year 300 as our testing dataset and the remaining characters from previous years are used for training the model. Thus our dataset contains a total of 83 and 1864 characters in testing and training data, respectively. We formulate the problem of character death prediction as a social network analysis and classification problem [6,8]. Figure 3 illustrates the high-level architecture of the proposed research framework. The proposed architecture is a multi-step process consisting of four major phases: feature importance, edge weight assignment in the social network of characters, death-prone score model, and prediction and evaluation. We discuss each of these phases in the following subsections:

3.1 Feature Importance

As discussed in Sect. 3, each character in the series has a variety of features describing the profile of each character. For example, house, culture, battle fought as an attacker or a defender, siblings, appearance in the book and many more. The same set of features defines the notion of similarity between two characters in the series. The existing studies assign equal weight to each attribute for computing the interactions between characters and ignore the fact of the relative

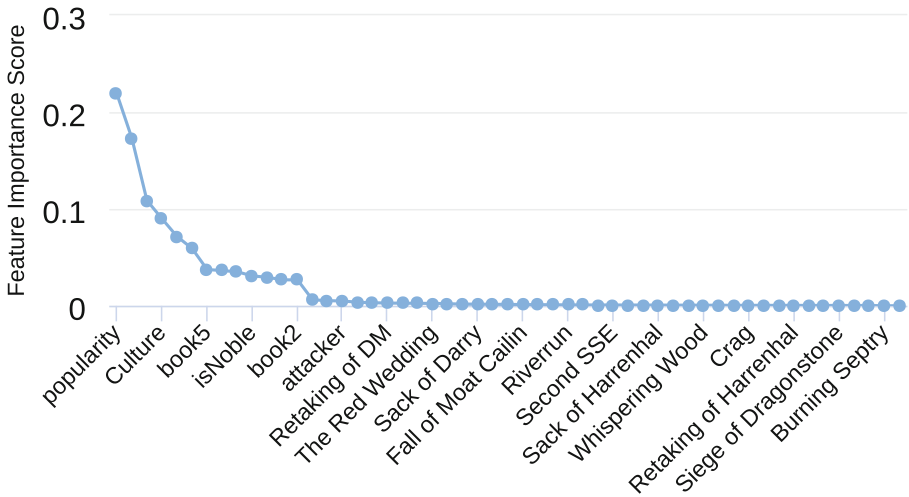

Fig. 4. Importance score of all features present in the dataset, arranged in the decreasing order of the score.

importance of different types of interactions. For example, the importance of belonging to the same *house* over fighting as opponents in the same *battle* or the *spouse* relation over belonging to different house or culture. In this paper, we propose to use the relative importance of each attribute and assign them weights to compute the similarity between the two characters. Each character in the dataset is represented as a vector in a k-dimensional space where k represents the number of attributes. The similar vector can be visualized as a Utility matrix where column represents the k attributes and rows represent the characters. The similarity computation is technically challenging due to a sparse utility matrix.

We compute the importance of each feature and assign a normalized weight to them. The tree-based classification, correlation, and leave-p-out methods are some of the most standard algorithms used for determining the importance of the features. We use ExtraTree Classifier- a tree-based ensemble method to compute the weight of each feature [7]. ExtraTree Classifier [1] focuses on randomizing both the attribute and split point for the tree node. We use ExtraTree Classifier as it outperforms Random Forest and Decision Tree-based methods by reducing computational complexity linked to the determination of optimal cut-points (tree pruning). Unlike bagging approach in the random forest algorithm, it finds the best fit among various random splits for the random set of attributes. The more randomness of the algorithm addresses the challenge of bias and minimizes the errors (less correlated) made by baseline models. Based on the feature importance calculated in our dataset (refer to Fig. 4) we find that *popularity*, *culture*, *house*, *isnoble*, role of an *attacker* in the battle, and appearance in *books number* 2 and 5 are some of the most significant attributes in the dataset.

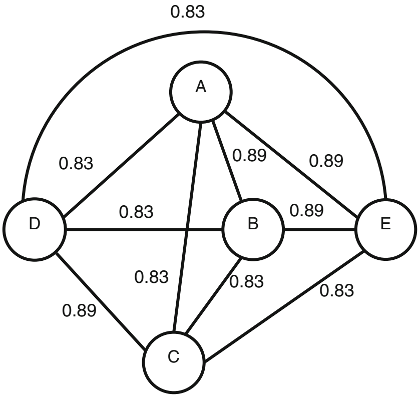

Fig. 5. A snapshot of top 10 weighted edges between 5 different characters in the social graph

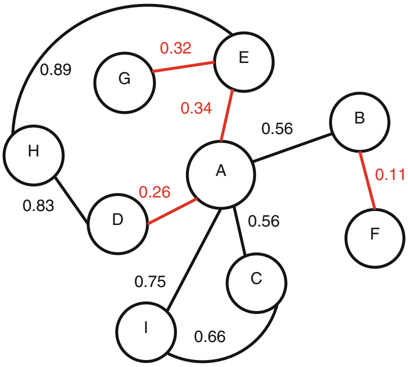

Fig. 6. A Snapshot of the discarded edges due to the relatively low similarity between characters

3.2 Edge Weight Assignment in the Social Network

The existing studies create a social network for different relations between characters to predict their death. However, for individual feature or relation, the predictions are made differently for the same character [2]. Further using one attribute for a network generates a combination of directed and undirected graphs for different relations. For example, for *house* or *culture* attributes, the graph has undirected edges with weakly connected components. While for *battle fought* as a relation, the graph has directed edges with many isolated nodes since not all characters participated in the battles. We extend the idea of creating a social network for one feature and instead represent the relation between two nodes as one single edge that depicts the similarity between two characters. In prior literature, all edges have an equal cost of 1. We, however, propose to assign different weights to each edge equal to the similarity between two nodes in the network. We create a character profile from the utility matrix [12] of all the features and their associated importance in the dataset. Thus each character is represented as a vector $u = [\alpha_1 A_1, \alpha_2 A_2, ..., \alpha_n A_n]$ where α_i represents the importance of attribute A_i. We take each character profile from the character-attribute utility matrix and use centered cosine similarity to compute the similarity every character pair. The centered cosine similarity addresses the challenge of zero-valued vectors (features of a character) and computes the similarity normalized across all attributes, thus keeping a check on the dominance of high magnitude- attributes over other attributes.

Figure 5 shows a concrete example of the edge weight computes between character pairs. Figure 5 illustrates that reciprocal links between only four characters ranked the highest in weights. We conduct a manual inspection on the dataset and observe the patterns in these characters (A: Stafford Lannister, B: Tybolt

Lannister, C: Damon Lannister (lord), D: Gerold Lannister, E: Daven Lannister). The pattern reveals that these characters belong to the same family, hold the similar titles and the popularity. Further, they have same gender, present in the first four books, and have participated in the battles together. Such a degree of similarities justifies the weights the edges between them hold. We further discard the links that have negligible similarity between two characters. The similarity computation is a function of type of interaction and the frequency of interaction in the book series. For example, two characters who fought multiple battles together. The higher the frequency of interaction increases the score of the edge weight. However, the edge weight further depends on the importance of the interaction. For example, two character having same gender is less important than the two different gender characters belonging to the same house. Thus, we expand the idea of weighted TF-IDF and exploit the application of ExtraTree Classifier and User Profile Utility Matrix to compute the edge weight. Some character pairs who have higher frequency but do not have high rated interactions are linked via less weight edges. To improve the performance of out model and avoid overfitting in prediction, we discard edges from the network consisting of negligible weights. To compute the threshold for removing edges, we compute the mean of all edge weights. Figure 6 shows the concrete examples of characters with very low similarity and hence not having a direct connecting edge in the network. For all the characters, we had approximately 1.8 million character pairs. We take only top 6000 similarity scores and plot their social network.

Figure 7 shows a snapshot (sample) of the character pairs present in our dataset. Figure 7 reveals that despite having a different culture, book or battle attributes, many characters end-up connected in the network due to the weighted feature importance and character similarity model. Since house and culture are two of the most significant attributes in the dataset, the majority of the characters in the sample network belong to the same house (Stark family). The width of the edge in the network represents the similarity score between two characters based on a variety of parameters such as house, battle, culture, and many more. Colour represents the clusters of characters having similar behaviour within parameters. The size of a node represents the number of interactions in series (degree of a node in the network).

3.3 The Proposed DPS Model

As discussed in Sect. 3, we divide our experimental dataset into two parts: all the characters died before year 300 are used as training nodes while other characters (died in the year 300 or still alive) are used as the testing nodes. To predict the death of characters present in the testing dataset, we propose to compute the death-prone score of each character and determine how likely a character is going to die next. We extend the idea of PageRank [5,13] and damping factor [3] to compute the Death-Prone Score (DPS) of each character. We propose to quantify the proneness of death by integrating the idea of PageRank with the proximity between dead and unknown/candidate characters in the social

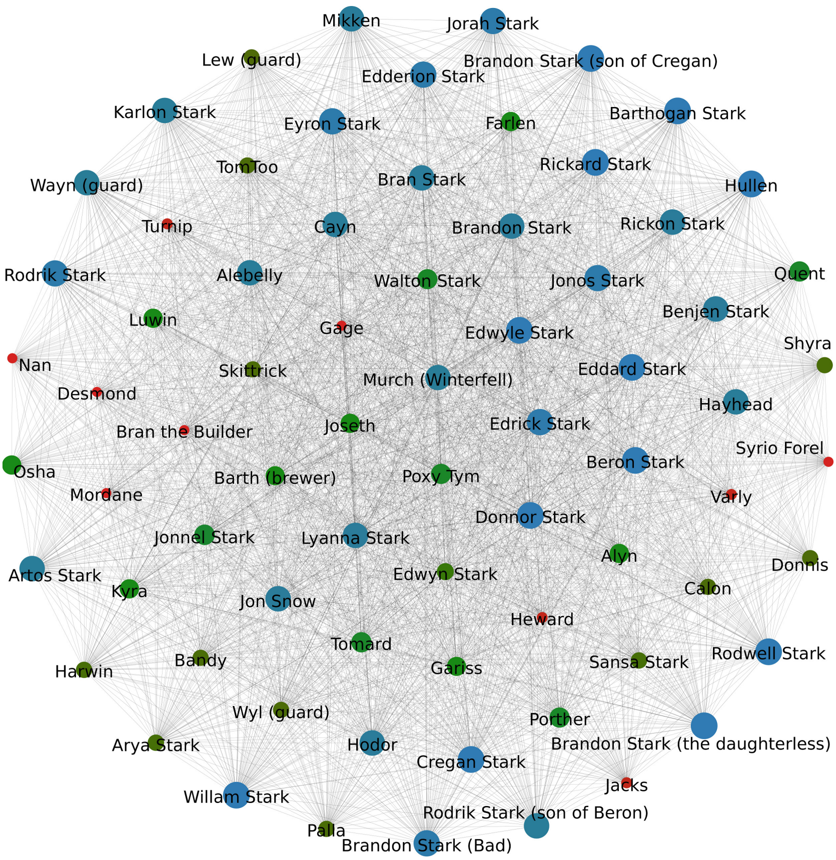

Fig. 7. A snapshot of social network graph created for various characters in the dataset. The width of the edge represents the weight of the overall interaction based on the priority of various interactions. (Color figure online)

network. We take each dead node (u) in the network present in the training dataset, and it's neighbouring candidate characters (v) are assigned a DPS of

$$DPS(v) = \frac{\alpha \times \sum_{i=0}^{N} \prod_{j=0}^{k-1} \frac{w(u_j, u_{j+1})}{2^j \times deg(u_{j+1})}}{N} \quad (1)$$

where $PR(u) = \sum_{x \in X} \frac{PR(x)}{deg(x)}$ denotes the PageRank of a node u. The PageRank value for a node u is dependent on the PageRank values for each node x contained in the set X (the set containing all nodes linking to node u), divided by the number $deg(x)$ of links from node x. In Eq. 1, k is the number of hops between the dead node $\{u\}$ and the candidate node $\{v\}$, the value of j represents the intermediate nodes, i represents the number of dead nodes (N) connected to v, and α represents the damping factor set to the standard value as 0.5. We divide

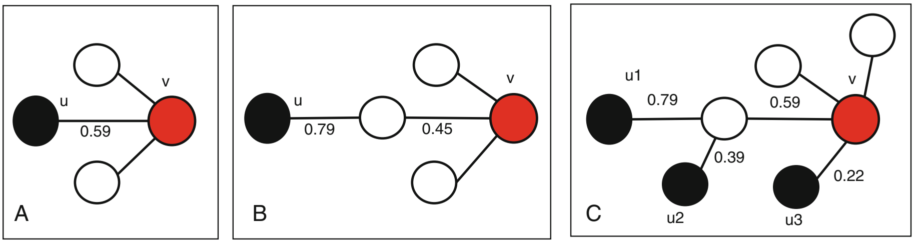

Fig. 8. Shows different scenarios of candidate characters (red node) to be connected with dead character (black node) for computing the death prone score in the social network (Color figure online)

the weighted score by $deg(v)$ and N to normalize the effect of many counts of neighbors of v.

Based on the closeness of a dead character (u) to the candidate character (v) in the network, we evaluate our model for the following possible scenarios:

1. **u and v are the adjacent nodes:** Figure 8 shows a snapshot of the network in which a dead node u (represented as black node) and testing character v (represented as red node) are one hop away from each other. In such a network, v might be connected to other nodes (represented as white nodes) as well. As shown in Fig. 8, the DPS of v strictly depends on the proximity and similarity to u. For the given similarity score in Fig. 8, we compute the DPS of v to proofread our proposed formula of DPS score. Since v is connected to only one dead node, the value of N is equal to 1. Further, the nodes are adjacent to each other hence, the value of k is 1. For such nodes, we compute the death-prone score of alive character as

$$DPS(v) = \alpha \times \frac{w(u,v)}{deg(v)} \tag{2}$$

   Given the 59% similarity between nodes u and v, the death-prone score of v is $0.5 \times 0.59/3 = 0.098 = 9.8\%$.
2. **u and v are multi-hop away:** Given a network where the dead node and the candidate node characters are two or multiple hops away (have one or multiple intermediate nodes in between), the $DPS(v)$ does not only depends on the similarity from the u but also depends on the proximity of intermediate nodes from u. Therefore, we use the joint probability measures and compute the $DPS(v)$ as a product of $DPS(u,x)$ and $DPS(x,v)$ where x represents the intermediate node. The product of DPS with the intermediate score signifies that as the closeness between a dead node and candidate node decreases the contribution of the trail of weights should also be decreasing as the number of intermediate nodes increase. For the given similarity scores in Fig. 8B, we compute the $DPS(v)$. For such a network, we simplify the formula as

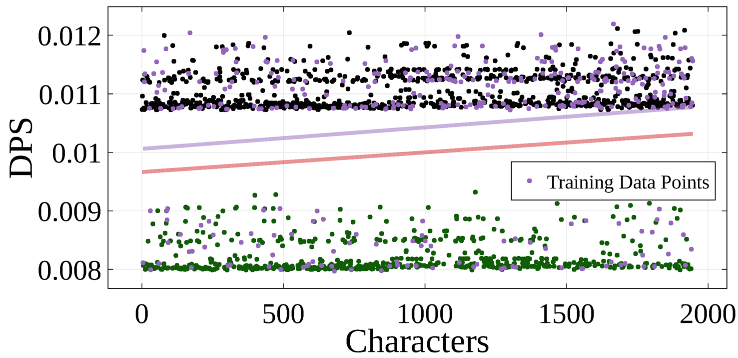

Fig. 9. Death Prone Score of each character in the dataset. Demonstrates the linearly separated characters based on their DPS value.

$$DPS(v) = \alpha \times \prod_{j=0}^{k-1} \frac{w(u_j, u_{j+1})}{2^j \times deg(u_{j+1})} \tag{3}$$

Given the similarity score of u and v with intermediate nodes (79% and 45%, respectively), the death prone score of v is 1.48% ($(0.79/2) \times (0.45/2 \times 3) \times 0.5 = 0.0148$) which is relatively lower than the DPS of intermediate node which turns out to be 19.8%.

3. **v has a connecting path from multiple dead nodes:** Considering a real-world scenario, we analyze the possibility of the network where a character can have connecting edges from multiple dead nodes and can also have intermediate nodes in between the dead node and itself. Figure 8C shows an example of such network. Similar to above points, in such case, the death prone score of candidate node is affected by the proximity of each connecting dead node and the intermediate node. For the intermediate nodes, we use the above formula, while for multiple connecting dead nodes, we use the weighted average of $DPS(v)$ computed explicitly w.r.t. each dead node computed using Eq. 3. Therefore, for such cases, we use our generalized formula proposed in Eq. 1. Given the similarity scores of v and intermediate node with various dead node characters, the death prone score of v is $((0.395 \times 0.073) + (0.195 \times 0.073) + (0.55) \times 0.5)/3 = 0.296/3 = 0.098 = 9.8\%$. While considering the intermediate node as our candidate node, the DPS turns out to be 32% which makes sense since the node is adjacent to more dead nodes (u_1 and u_2) than v and has higher character similarity.

3.4 Classification and Results

Figure 9 shows the death-prone score of all the characters present in our dataset and reveals that there is a fair discriminatory line separating the characters with low DPS from the characters with high DPS value. Since the data points are

Table 1. Confusion Matrix and prediction results for alive and dead characters present in the testing dataset

		Predicted	
		Alive	Dead
Actual	Alive	TP = 1138	FN = 313
	Dead	FP = 52	TN = 85

Metrics	Accuracy
Precision	95.6%
Recall	78.4%
F-score	86.21%

linearly separable, we find the least square fit in the dataset (optimal linear parameters) and draw the line separating data points. We apply the simple rule-based classifier to label each character. All the characters below margin are predicted alive while the characters with a DPS appearing above margin line are labeled dead in the year 300. We check the performance of our proposed methodology and DPS model against the ground truth available for year 300 data. We use the standard measures in information retrieval and report the accuracy in the form of precision and recall. Table 1 shows the confusion matrix for the classification and prediction results. Table 1 reveals that among 1588 characters present in our testing dataset, 85 are correctly identified as dead while 1138 are correctly predicted as to be alive in the upcoming year or book. The model, however, reports a misclassification of 37% and 21% in predicting dead and alive characters, respectively. The proposed model achieves a precision of 95.6 and a recall of 78.4%. We further compute the f-score of our model and achieves an accuracy of 86.21%.

4 Conclusions and Future Work

In this paper, we exploit the application of social network analysis for the prediction of dead and alive characters in the novel book series named "A game of thrones". We address the challenge of the vast domain of features and sparsity in the dataset by assigning a weighted score to each feature referred to as feature importance. We automate the interpretation of the relationship between characters by computing the weighted similarity between every character pair in character-features utility matrix. We create a social network of all characters present in our dataset and propose a network measure based model to predict the characters that are highly likely to die or stay alive in upcoming chapters or novel in the series. We propose a Death-Prone Score model that takes proximity and similarity between characters as inputs and generate a score. We further use this score to predict the death of a character. We also test our model for various situations of a candidate character in the network such as direct proximity with a dead node (character), two or multi-hop away from a dead node, and connected with multiple dead nodes. The idea is to improve the performance of our prediction model by giving more weightage to the characters having multiple links with a dead node. Our results show that computing feature importance normalizes the factor of biases towards character profile and house, are the most discriminatory and significant attributes in the dataset to find the character similarity in GoT book series. Further, unlike the different proximity score for each relation, an aggregated weight can be assigned to each node pair in the network that is an efficient approach to find similar characters in the series despite the lesser number of interactions and direct similarity. Our results reveal that the proposed DPS model achieved an accuracy of 86.21% while reporting precision and recall of 95.6% and 78.4%, respectively.

Future work includes performing an event-based analysis for the prediction of the character's death. The work presented in this paper predicts the death of a character based on the static analysis of year 300. We plan to extend the analysis and forecast the death of character B in the testing dataset when character A has died from the same year and thus capture the dynamic changes in the network. Our future work also includes training a supervised classifier model to automatically predict the characters' death based on their metadata and death prone score model. Computing death-prone score is not limited to predicting the death of characters but has its application in a variety of domains. The future work includes testing the model for cellular and femtocell network architecture. The aim is to capture the devices with similar configuration and predict the next devices to be attacked in case of an earlier malicious attack happened on one or more devices.

References

1. Athey, S., Tibshirani, J., Wager, S., et al.: Generalized random forests. Ann. Stat. **47**(2), 1148–1178 (2019)
2. Beveridge, A., Shan, J.: Network of thrones. Math Horiz. **23**(4), 18–22 (2016)
3. Boldi, P., Santini, M., Vigna, S.: PageRank as a function of the damping factor. In: Proceedings of the 14th International Conference on World Wide Web, pp. 557–566. ACM, New York (2005)
4. Bonato, A., D'Angelo, D.R., Elenberg, E.R., Gleich, D.F., Hou, Y.: Mining and modeling character networks. In: Bonato, A., Graham, F.C., Prałat, P. (eds.) WAW 2016. LNCS, vol. 10088, pp. 100–114. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49787-7_9
5. Chakrabarti, S.: Dynamic personalized pagerank in entity-relation graphs. In: Proceedings of the 16th International Conference on WWW, pp. 571–580 (2007)
6. Farine, D.R.: When to choose dynamic vs. static social network analysis. J. Anim. Ecol. **87**(1), 128–138 (2018)
7. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
8. Haghani, S., Keyvanpour, M.R.: A systemic analysis of link prediction in social network. Artif. Intell. Rev. **52**(3), 1961–1995 (2019)
9. Ibnoulouafi, A., Haziti, M.E., Cherifi, H.: M-centrality: identifying key nodes based on global position and local degree variation. J. Stat. Mech.: Theory Exp. **2018**(7), 073407 (2018). http://stacks.iop.org/1742-5468/2018/i=7/a=073407
10. Janosov, M.: Network science predicts who dies next in game of thrones. Department of Network and Data Science Group at Central European University (2017). https://networkdatascience.ceu.edu/article/2017-07-08/network-science-predicts-who-dies-next-game-thrones

11. Jia, R., Liang, P.: A game of thrones: power structure and the stability of regimes. In: Wärneryd, K. (ed.) The Economics of Conflict: Theory and Empirical Evidence. CESifo Seminar Series, pp. 79–104. MIT Press, Cambridge (2014)
12. Lakiotaki, K., Delias, P., Sakkalis, V., Matsatsinis, N.F.: User profiling based on multi-criteria analysis: the role of utility functions. Oper. Res. Int. J. **9**(1), 3–16 (2009)
13. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007)
14. Maurella, C., et al.: Social network analysis and risk assessment: an example of introducing an exotic animal disease in Italy. Microb. Risk Anal. **13** (2019). https://doi.org/10.1016/j.mran.2019.04.001. ISSN: 2352-3522
15. Moradabadi, B., Meybodi, M.R.: Link prediction in weighted social networks using learning automata. Eng. Appl. Artif. Intell. **70**, 16–24 (2018)
16. Pierce, E., Kahle, B., Downey, A.: Bayesian survival analysis for "game of thrones". A blog by Allen Downey (2015). http://allendowney.blogspot.com/2015/03/bayesian-survival-analysis-for-game-of.html
17. Sah, P., Mann, J., Bansal, S.: Disease implications of animal social network structure: a synthesis across social systems. J. Anim. Ecol. **87**(3), 546–558 (2018)
18. Schubart, R., Gjelsvik, A.: Women of Ice and Fire: Gender, Game of Thrones and Multiple Media Engagements. Bloomsbury, New York (2016)
19. Shurkin, J.: Using social networks to analyze the classics. Inside Science (2012). https://www.insidescience.org/news/using-social-networks-analyze-classics
20. Tracy, P.: Using big data to accurately predict death in the 'game of thrones' series. Intelligence on all things wireless, RCR Wireless News (2016). https://www.rcrwireless.com/20160915/big-data-analytics/game-of-thrones-tag31-tag99
21. Tschiatschek, S., Singla, A., Gomez Rodriguez, M., Merchant, A., Krause, A.: Fake news detection in social networks via crowd signals. In: Companion Proceedings of the The Web Conference 2018, pp. 517–524. International World Wide Web Conferences Steering Committee (2018)
22. Zhang, Y., Mu, L., Shen, G., Yu, Y., Han, C.: Fault diagnosis strategy of CNC machine tools based on cascading failure. J. Intell. Manuf. **30**(5), 2193–2202 (2019)