A Hybrid TF-IDF and RNN Model for Multi-label Classification of the Deep and Dark Web

Ashwini Dalvi[1], Soham Bhoir[2], Nishavak Naik[3], Atharva Kitkaru[4], Irfan Siddavatam[5], Sunil Bhirud[6]
Department of Computer Engg., Veermata Jijabai Technological Institute, Mumbai, India[1,6]
Department of Information Technology-KJSCE, Somaiya Vidyavihar University, Vidyavihar, India[2,3,4,5]

Abstract—**The classification of content on the deep and dark web has been a topic of interest for researchers. Researchers focus on adopting more efficient and effective classification methods as the data available on deep and dark web platforms continues to grow. Multi-label classification is the approach for simultaneously categorizing content into multiple classes. To address this, a hybrid approach combining Term Frequency-Inverse Document Frequency (TF-IDF) and Recurrent Neural Network (RNN) has been proposed. The approach involves preprocessing a dataset of Hypertext Markup Language (HTML) documents, selecting specific HTML tags to generate embeddings using TF-IDF, and using an RNN model for multi-label classification. The proposed model was evaluated against commonly used methods (Binary Relevance, Classifier Chains, and Label Powerset) using precision, recall, and F1-score as evaluation metrics, demonstrating promising results in accurately classifying data from the deep and dark web. This contribution represents a noteworthy advancement for researchers and analysts working in this field.**

***Keywords*—Deep web; dark web; multi-label classification; TF-IDF; FastText; RNN**

I. Introduction

A deep web is a portion of the internet not accessible by traditional search engines. A subsection of the deep web, the dark web, is known for its anonymity and association with illegal activities, while it requires specialized software to access [1]. An encrypted network called Tor, or "onion routing," is often used to access the dark web like the Tor browser. The Tor Network, a free, open-source software platform, enables anonymous communication. To provide anonymity to its users, Tor encrypts their internet traffic before it reaches its final destination and bounces it around multiple servers.

To comprehend cyber threat intelligence from the dark web, researchers collect deep and dark web data using web crawlers. Deep web crawler collects data from websites not listed in traditional search engines. Deep web data collection aims to find resources, often valuable, and provide insight into various domains, including science, medicine, and finance. Databases and APIs are examples of deep web content that may contain structured data suitable for data integration. In addition, researchers can use deep web data to study the behavior of users in online communities and analyze trends in e-commerce [2]. It is essential to use deep web data from vast and diverse corners of the internet that are difficult to access conventionally. Text classification is the primary method for inferring information from deep and dark web pages crawled to analyze content. Using text features, researchers propose a method to classify Deep Web data sources [3].

Since the dark web is anonymous, criminals can engage in these activities without fear of being caught, causing a proliferation of criminal enterprises. As a result, many illegal activities are happening on the dark web, including drug trafficking, arms trafficking, human trafficking, and selling stolen data. However, a challenge is associated with studying the dark web because of its unique characteristics [4]. First, due to the encrypted nature of the dark web, identifying users and tracking data origins is very difficult. Consequently, tracing the source of criminal activities and cyber threats is challenging for law enforcement agencies and researchers.

The dark web has been investigated for its content in recent years. Law enforcement agencies and cybersecurity experts can detect and prevent criminal activities using machine learning approaches on the dark web, where there is vast data. Researchers discuss how the dark web information they find can be assessed for its relevance, usefulness, and appropriateness [5]. Researchers have used Machine Learning (ML) approaches to investigate content on the dark web. Researchers can better understand how activities on the dark web are patterned by analyzing individual use cases and different forms of data. For example, text and image data can be automatically classified and analyzed using algorithms to detect potential illicit activities, such as illegally selling goods and services [6]–[8].

Using machine learning algorithms and external knowledge sources maximizes the efficiency and effectiveness of identifying and categorizing deep and dark web content. However, the lack of access to these deep and dark webs makes research into their nature and structure challenging. One standard method is to manually label crawled sites per a series of categories and then use this corpus as a training corpus for automated crawling in the future. While this approach has its benefits, it has limitations because it is time-consuming, expensive, and valuable only for specialized tasks. Also, related research mainly focuses on classifying content with a single label. As per the authors' knowledge, the presented work is the first attempt to label deep and dark web content with multi-label classification.

The contribution of this study lies in the methodology proposed, which provides a comprehensive and systematic approach for labeling hidden content on the deep and dark web with multi-label. This approach starts with a dataset of HTML documents scraped from the deep and dark web. Then, as part of the preprocessing, specific HTML tags are selected, irrelevant characters are removed, and embeddings are generated using TF-IDF. In the next step, FastText assigns labels to documents according to their similarity. Text classification and language modeling can be accomplished with FastText, a popular machine-learning algorithm.

Recurrent Neural Networks (RNNs) are trained on pre-processed datasets after the documents have been labeled. In particular, RNNs are well suited to dealing with sequential data, such as text. This study compared the proposed method to three existing methods: Binary Relevance (BR), Classifier Chains (CC), and Label Powerset (LP). Precision, recall, and F1-Score are used as evaluation metrics to measure the performance of the proposed approach. The proposed approach for multi-label classification of deep and dark web content has significant implications for enhancing security and combatting criminal activities on these platforms.

The following paper is organized into four sections: Related Work, Proposed Approach, Results and Discussion, and Conclusion. Section II, related work covers the need for deep and dark web content classification and existing multi-label classification approaches. Section III, proposed approach explains the methodology used, while Section IV, Results and Discussion, presents the findings. As a final section of the paper, Section V, the conclusion section summarizes and suggests future research directions.

II. Related Work

Information obtained from deep web sources often lacks structure, making it challenging to categorize and analyze. Instead, specialized algorithms must be used to extract text features to classify web pages in this vast and complex world of the deep web. As part of their extraction process, these algorithms use various techniques, including keyword encoding, topic modeling, sentiment analysis, and entity recognition, to identify meaningful features in the text. The algorithms can extract valuable insight from the text by categorizing it based on the extracted features. For example, in the context of mobile app stores, the paper highlights the importance of text feature classification algorithms for collecting and analyzing deep web data [9].

Deep web crawlers use their classification modules to understand the context of the data they collect. For example, through text classification algorithms, the classification module provides insight into the actual content of a web-page. The researchers proposed an Accurate crawler to harvest deep web content with accurate classification [10]. By ranking sites based on the similarity of their content, the framework attempted to reduce the number of pages that need to be visited. Consequently, deep web content can be extracted and classified more accurately. In addition, researchers proposed a smart crawler to search the deep web efficiently while avoiding irrelevant pages [11]. Dark web content classification is necessary because these parts of the internet are often used for illegal and criminal activities, like drug trafficking, drug sales, and cybercrime. Natural language processing techniques are used to classify texts on the dark web, and supervised machine learning algorithms are used to classify dark web content.

Researchers proposed an Automated Tool for Onion Labeling (ATOL) for mapping dark web content to thematic labels [12].In this approach, popularity scores for different categories were calculated using TF-IDF. Three components comprised the ATOL system: a module that finds keywords for different categories, a classification framework that maps onion content to categories when labeled data is available, and a clustering framework that categorizes onion content using external knowledge sources when labeled data is lacking.

The researchers proposed a crawler to search dark web links for markets [13]. A dataset was created and pre-processed using data-cleaning techniques. Linear Support Vector Machines outperformed Random Forests and Nave Bayes in classifying dark web pages. The proposed system effectively identified and classified dark web pages with high accuracy, precision, recall, and F1 score. In addition, researchers proposed a modified frequency-based term weighting scheme for identifying dark web content [14]. In a dataset selected from the dark web Portal Forum, the proposed term weighting scheme was compared to Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IFD), and Term Frequency-Relative Frequency (TF.RF). Experimental results demonstrate that the proposed scheme outperforms other term weighting techniques based on classification accuracy and other evaluation measures.

Researchers analyzed using the Vector Space Model with two-term weighting schemas: TF and TF-IDF, to determine some of the most discussed topics within Arabic dark web forums [15]. In addition, researchers proposed a new method of tagging extracted data to provide a more balanced and effective method for detecting text features [16]. ELEMENT, a modified TF-IDF algorithm, considered several factors, including document length and term length. One study applied the TF-IDF to classify dark web marketplaces according to their offered products [17]. In addition, researchers identified language characteristics of dark web forums related to drug markets using TF-IDF [18]. Researchers confirmed the similarity between the subjects of dark web forums with affect analysis [19]. Focused topics identified in forums were piracy, hacking, drugs, politics, revolution, weapons, and guns.

Researchers evaluated some text embeddings and classifiers for classification tasks in the darknet domain [20]. In the study, researchers compared text classification to keyword searches by training the classifier using keyword search results. The study concluded that text classification performed better than a keyword-based search. Although dark web content is typically classified using a single label, multiple-label classifications can be helpful in cases where content belongs to multiple categories. However, dark web websites or contents are typically categorized into one of several categories, such as drugs, weapons, or hacking. Therefore, multi-label classification is helpful for dark web content that may be classified with multiple labels. For example, dark web data can be analyzed for multi-label classification tasks based on multiple labels such as ‘fraud,’ ‘drugs,’ and ‘weapon Researchers reviewed three significant areas of multi-label learning: paradigm formalization, learning algorithms, and learning environments [21]. Besides defining multi-label learning and evaluating its outcomes, the research examined traditional and deep learning-based algorithms for multi-label learning. The empirical results also discussed how different learning settings, such as feature selection and transfer learning, affect the learning process and algorithm performance.

Multi-label classification, in combination with other machine learning tasks, has led to the development of two main categories of methods: Problem Transformation (PT) and algorithm adaptation [22]. Several PT methods are independent of the algorithm and involve transforming a multi-label classification task into a single-label classification, regression, or ranking task. Methods such as BR, CC and LP fall into this category. The review provided valuable insights and techniques for multi-label learning, but it is necessary to adapt further and study these techniques for dark web data classification.

The presented work has been proposed to classify dark web data using techniques such as BR, CC, and LP. By including label interdependence and using probabilistic models to predict label combinations, these methods can improve the accuracy of multi-label classification.

BR is a decomposition method that assumes labels are independent and trains binary classifiers separately to determine the Relevance of each label [23], [24]. It has been proven that BR produces good machine learning classifiers both computationally and as a result of several metrics. An example of the application of BR is covered in the study, which developed a novel method for multi-label text classification for Arabic texts using binary relevance [25]. The study examined five multi-label classification approaches for enhancing Arabic multi-label text classification, including Support Vector Machines (SVMs), k-Nearest Neighbors (KNNs), Naive Bayes (NBs), and different classifiers.

Aside from BR, other multi-label classification models have also been developed to deal with label dependency. As one example, the CC method is becoming increasingly popular due to its simplicity and promising results [26]. As a method for solving multi-label learning problems, classifier chains consist of chaining together off-the-shelf binary classifiers in a directed structure [27]. Then, the label predictions are used to refine other classifiers. Various datasets and evaluation metrics have been used to evaluate the effectiveness and flexibility of this method.

There has been an alternative method developed to address label dependency in a multi-label classification called LP. The LP transformation aims to transform multi-label learning problems into single-label multi-class problems by transforming these into one-label multi-class problems [28]. By using LP transformations, all label combinations present in the original dataset are transformed into one label.

Multi-label classification has gained attention recently due to its application in solving complex real-world problems [29]. For example, multi-label classification is helpful in text classification, image annotation, and bio-informatics scenarios with multiple labels associated with each instance. In addition, using multi-label classification and feature selection techniques is an upcoming approach to identifying the most relevant features. Selecting features aims to reduce the number of irrelevant features while keeping only the most informative ones. Removing noisy or irrelevant features can enhance the performance of multi-label classification models by reducing the input space dimension.

Using multi-label classification for multitask learning is another example of combining multi-label classification with other machine learning tasks. A multitask learning model uses a single model to learn multiple related tasks simultaneously. However, as each document may contain multiple correlated labels, it is challenging to classify text using multiple labels. Therefore, the researchers introduced a new multi-label text classification technique, learning feature combinations from documents and labels while reinforcing label correlations [30]. As a result, researchers avoid label order dependency and ensure that label correlations are effectively learned. Applying the multi-label text classification technique for dark web content is feasible. The dark web contains a wide range of diverse and often hidden content, making its identification and classification particularly challenging. However, with the multi-label text classification approach, the model can learn to capture correlations between labels, even if they were not explicitly observed in the training data. This enables the possibility of improving dark web multi-label classification.

The proposed study presents a comprehensive and systematic approach to multi-label classification of dark web text. The method involves several steps: first, collecting HTML documents from the dark web, then preprocessing them by selecting relevant content, and generating embeddings using the TF-IDF method. Next, labels are assigned to the documents based on their similarities using FastText. Finally, an RNN is trained using these labels. The model's performance for multi-label classification problems is evaluated using Precision, Recall, and F1-Score. Section III provides a detailed description of the proposed approach.

III. Proposed Approach

The following discussion covers the proposed procedure for selectively picking and labeling the content of deep and dark web HTML tags based on their frequency and Relevance using the TF-IDF and FastText embeddings. To multi-label a text document, the authors propose combining TF-IDF and FastText. The TF-IDF statistic evaluates the importance of a term in a document by considering its frequency and inverse document frequency [31]. FastText, on the other hand, is a neural network-based approach that captures the semantic meaning of words through word embeddings [32].

The proposed method extracts TF-IDF features from textual data and trains a FastText model on top of these features. The trained model can then predict multiple labels for a single text instance. This method has yielded promising results in various multi-label classification tasks, including text categorization and sentiment analysis. The combination of TF-IDF and FastText yields a robust and efficient solution for multi-label textual data classification. The authors proposed neural network architecture for binary classification. The proposed architecture is a feed-forward neural network, a Multi-Layer Perceptron (MLP). The network is implemented using the Sequential API, which allows for the easy creation and addition of layers in a linear stack. Fig. 1 illustrates the model architecture along with crucial details regarding layer activations and layer names. The diagram provides a visual representation of the network's structure, showcasing the flow of information through the layers and the respective labels assigned to each layer.

The architecture consists of two dense layers. A dense layer, also known as a fully connected layer, is one in which every neuron in the layer is connected to every neuron in the previous layer. The first dense layer has 50 units, an

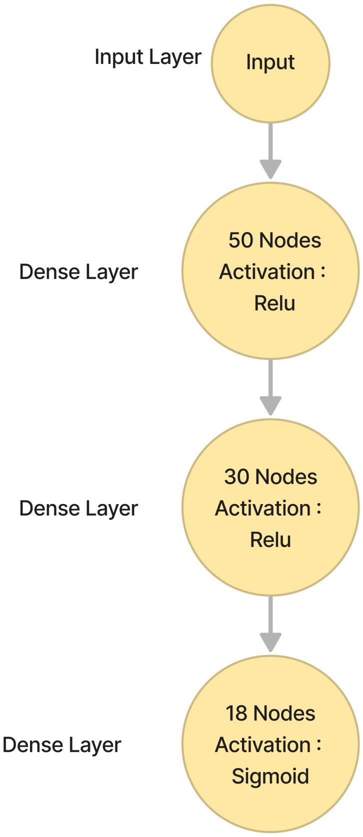

Fig. 1. Model architecture of neural network.

Algorithm 1: Multi-Label Text Classification Algorithm

```
Input  : $X$ : Text data
Input  : $Y$ : Labels
Procedure MultiLabelTextClassification
    $X, Y$
$X_{\text{preprocessed}} \leftarrow$ preprocessing($X$) ;
    // Cleaning and tokenizing the text data
$X_{\text{features}} \leftarrow$ feature_extraction($X_{\text{preprocessed}}$) ;
    // Extracting relevant features
$tf(i,j) \leftarrow \frac{n_{i,j}}{\sum_k^V n_{k,j}}$ ;    // Calculating term frequency
$idf(i) \leftarrow \log \frac{N}{\sum_{j=1}^N [w_i \in d_j]}$ ;    // Calculating inverse document frequency
$X_{\text{tf-idf}} \leftarrow tf(i,j) \cdot idf(i)$ ;    // Calculating tf-idf representation
$\mathbf{Z} \leftarrow \mathbf{W} \cdot \mathbf{X}_{\text{tf-idf}} + \mathbf{b}$ ;  // Calculating linear transformation
$\mathbf{A} \leftarrow f(\mathbf{Z})$ ;    // Calculating activation function output
$\hat{\mathbf{Y}} \leftarrow$ softmax($\mathbf{A}$) ;    // Calculating predicted labels using softmax
$\mathbf{L} \leftarrow (Y, \hat{\mathbf{Y}})$ ;    // Calculating loss
$\theta \leftarrow \theta - \eta \cdot \frac{\partial \mathbf{L}}{\partial \theta}$ ;   // Updating weights and biases using backpropagation. Here, $\theta$ represents current weights and biases.
Output: $\hat{\mathbf{Y}}$ : Predicted labels
```

input dimension of 300, and a Rectified Linear Unit (ReLU) activation function. The output of the sigmoid function, when applied to the input features, will be a value between 0 and 1, which can be interpreted as the probability of the input features belonging to the positive class. The network is then compiled using the Adam optimizer, a binary cross-entropy loss function, and accuracy as the evaluation metric. Adam is a stochastic gradient descent optimizer that adapts the learning rate for each parameter. It is computationally efficient and has been demonstrated to work well in various tasks. The binary cross-entropy loss function is a loss function that is often used for classifying things into two groups. It measures the dissimilarity between the predicted probability distribution and the true distribution.

A. *Proposed Algorithm*

The authors proposed an algorithm for multi-label text classification of dark web data. Algorithm 1 describes the proposed algorithm.

Multi-Label Text Classification Algorithm

Further authors describe each step in the algorithm in the following discussion.

1) Data cleaning: Three main steps are involved in cleaning and preprocessing HTML tags' content. The first step is the selection of relevant HTML tags. A second step involves removing accented and non-ASCII characters, stopwords, languages other than English, and punctuation, which add little value to the content and may cause errors. The final process standardizes and normalizes the data by converting all the remaining content to lowercase and separating it into individual tokens and words. As a result, these steps enable further analysis of the HTML tags' content.

2) Performing TF-IDF on the data: The next step of the algorithm is to perform the TF-IDF on the data. The TF-IDF is calculated as the product of the TF, which is the number of times a token appears in a document, and the IDF, which is the logarithm of the ratio of the number of documents in the corpus to the number of documents containing the token.

The TF-IDF assigns higher weights to the tokens that frequently occur in a document but infrequently in the corpus and lower weights to the tokens that frequently occur in the corpus but infrequently in a document. The TF-IDF can help filter out the common or irrelevant tokens and highlight the specific or distinctive ones.

3) Picking tokens having the highest TF-IDF weights in the document: predefined categories or domains to which the researcher wants to assign the picked tokens based on their meaning or context. The generalized tokens may be selected or created by the researcher based on the research question, the scope, or the analysis criteria. The similarity is measured using a distance or a similarity metric, such as Cosine similarity or Euclidean distance, between the embeddings of the picked token and the generalized token. The similarity reflects the degree or the probability of the picked token belonging to the generalized token.

The further step is to pick the tokens having the highest TF-IDF weights in the document based on the selection criteria. The selection criteria are designed to balance the tokens' importance or Relevance and the diversity of representativeness of the tokens. The selection criteria consist of three cases:

- If more than twenty tokens are in the document, pick the top 60% of the tokens, sorted by the TF-IDF weights in descending order.
- If in the document, tokens are between six and twenty, pick the top 70% of the tokens, sorted by the TF-IDF weights in descending order.
- If there are fewer than six tokens in the document, include all the tokens.

The authors use the selection criteria to identify the most relevant and significant tokens for the analysis while avoiding redundant data and over-representation. Also, the selection criteria aim to maintain the overall context and subtleties of the document while capturing its primary or specific aspects.

4) Using FastText to generate embeddings of the picked tokens from the document: The next step of the algorithm is to use FastText, a library for efficient text classification and representation learning, to generate the embeddings of the picked tokens from the document. The embeddings are dense, low-dimensional, continuous, and real-valued vectors representing the tokens' semantics or meaning in a multi-dimensional space. The embeddings are trained or learned from a large dataset using a supervised or unsupervised learning algorithm, such as skip-gram or CBOW. The embeddings capture the tokens' co-occurrence or context and the relationships or similarities between the tokens. The embeddings can be used to measure the similarity or the distance between the tokens or to classify or cluster the tokens.

5) Calculating the similarity of every picked token with every generalized token from the list: The next step of the algorithm is to calculate the similarity of every picked token with every generalized token from the list using the embeddings. The generalized tokens are the predefined categories or domains to which the researcher wants to assign the picked tokens based on their meaning or context. The generalized tokens may be selected or created by the researcher based on the research question, the scope, or the analysis criteria. The similarity is measured using a distance or a similarity metric, such as Cosine similarity or Euclidean distance, between the embeddings of the picked token and the generalized token. The similarity reflects the degree or the probability of the picked token belonging to the generalized token.

6) Assigning the generalized token with the highest similarity score to the token: The further step is to assign the generalized token with the highest similarity score to the token based on the assignment criteria. The assignment criteria are designed to ensure the assignment's reliability or confidence and to handle exceptions or special cases. The assignment criteria consist of two cases:

- Only assign the generalized token to the picked token, if the similarity is more than 60
- Manually assign certain labels, such as "onion" or "tor", to the generalized token "network" if the picked token belongs to those labels. The assignment criteria aim to assign the picked token to the most suitable or the most likely generalized token while avoiding the assignment's errors or ambiguities. The assignment criteria also aim to respect the dark web's particularities or conventions and avoid misusing or abusing the generalized tokens.

7) Creating a set of assigned generalized tokens per document: The next step is to create a set of assigned generalized tokens per document by collecting the generalized tokens assigned to the picked tokens in the document. The set of assigned generalized tokens per document represents the labels or the topics of the document. It can be used as the input or the output of a classification task, such as topic modeling or sentiment analysis, or as the input or the output of a recommendation or search system. The set of assigned generalized tokens per document reflects the document's main or specific aspects or themes and the context or nuance of the document.

8) Splitting the dataset into train and test sets: The dataset has split into training and testing sets using a predetermined ratio or a sampling method. The training set is used to fit or train the classifier neural network model, and the testing set is used to evaluate or test the performance of the classifier neural network model. The ratio of the split method should be chosen based on the dataset's size, quality, or representativeness and the purpose or objective of the analysis. In this paper, authors used a ratio of 25% testing set and 75% training set.

9) Fitting classifier neural network model: Next, a neural network model is trained using the training set using generalized tokens as labels. The classifier neural network model is an artificial neural network used to classify or predict the labels of the input data based on its patterns or features. The classifier neural network model consists of an input layer, two hidden layers, and an output layer.

To train the classifier neural network model, the following hyperparameters are used:

- Sequential Classifier: The neural network architecture is designed in a sequential manner, where each layer is added one after the other.
- Activation Functions:
  - ReLU (Rectified Linear Unit): Used as the activation function for the hidden layers to introduce non-linearity into the model.
  - Sigmoid: Used as the activation function for the output layer in multilabel classification to produce probabilities for each label independently.
- Optimizer: The Adam optimizer is employed to minimize the loss function during training. Adam is a popular optimization algorithm that combines the benefits of both AdaGrad and RMSProp.
- Loss Function: Binary Cross-entropy is utilized as the loss function for multilabel classification. This loss function is well-suited for problems where each input

sample can belong to multiple classes, as is the case in multilabel classification tasks.

- Hidden Layers: The classifier neural network model includes two hidden layers, which contribute to the network's ability to learn complex representations from the data.
- Learning Rate: The learning rate is set to 0.004. The learning rate is a hyperparameter that determines the step size at each iteration of the optimization algorithm. A higher learning rate can result in faster convergence but may risk overshooting the optimal solution, while a lower learning rate can provide more stability during training but might take longer to converge.

During the training process, the classifier neural network model learns the mapping or relationship between the input data and the output labels by adjusting the weights or biases of the connections between the layers. This allows the model to handle the complexity or variability of the data and generalize or adapt to new or unknown data.

10)Evaluating the model using the test set: The twelfth and final step of the procedure is to evaluate or test the performance of the classifier neural network model using the testing set and the generalized tokens assigned as the labels. The evaluation is performed by comparing the predicted labels of the model with the actual labels of the testing set, using evaluation metrics such as accuracy, precision, recall, or F1 score. The evaluation metrics measure the quality or the reliability of the model's predictions and provide insights or feedback for the improvement or optimization of the model.

The algorithm creates a clean HTML document by cleaning and preprocessing the HTML tags' content. On the data, TF-IDF is performed. TF-IDF weights are used to select tokens. The algorithm generates a FastText embedding. Based on a list of generalized tokens, it calculates the similarity between each token. This algorithm assigns the token to the generalized token with the highest similarity score. Per the document, it creates a set of generalized tokens. Train and test sets are separated in the dataset. Algorithms are designed to extract the main or specific elements of a document.

Further, Section IV presents the result and discussion of the proposed algorithm.

IV. Result and Discussion

A. Dataset Description

For a proposed work, data was gathered using a custom deep and dark web crawler. In order to explore Tor's hidden services, the crawler used seeds provided by the Hidden Wiki page. The crawler continued to collect data as it encountered new links. An extensive deep and dark web dataset was collected as a result. The dataset comprised fifty thousand web pages from the deep and dark web. Because the dataset included HTML code for every web page, the file size increased significantly. The crawled files were cleaned by removing HTML tags, JavaScript, and non-English web pages. After the keywords have been extracted from each website, the cosine similarity between these keywords and a set of custom keywords has been calculated. The predefined labels in the dataset are Business, Cybersecurity, Education, Entertainment, Finance, Food, Health, Literature, Nature, Network, Politics, Security, and Shopping.

The evaluation phase includes the examination of model accuracy and loss graphs, which are presented in Fig. 2 and Fig. 3, respectively. These graphs serve as important tools to assess the model's performance and identify potential areas for improvement.

Fig. 2 and Fig. 3 illustrate the model's performance over time, visually representing accuracy and loss metrics. By closely monitoring these metrics, adjustments can be made to address issues such as overfitting or underfitting, ensuring the model's optimal performance and generalization to unseen data.

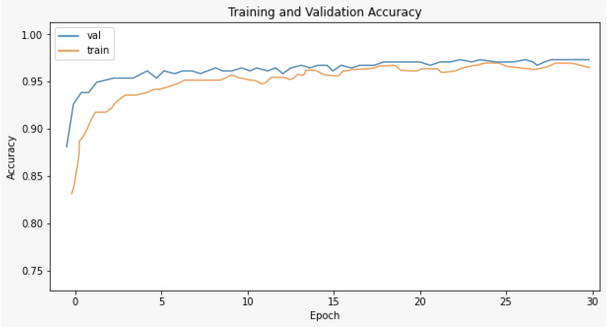

Fig. 2. Model accuracy graph for the proposed algorithm.

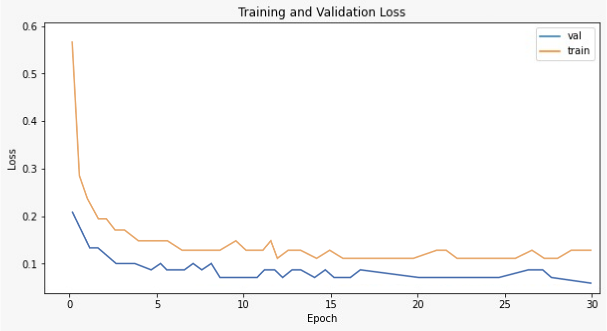

Fig. 3. Model loss graph for the proposed algorithm.

B. The Empirical Study of Multi-label Classification with BR, CC and LP Algorithms

Empirical studies collect and analyze data to gain insights and draw conclusions. The performance of different algorithms can be compared in the context of multi-label classification of dark web data, and the best algorithm can be determined through empirical research. In the context of multi-label classification of dark web data, an empirical study compares the performance of various algorithms such as BR, CC, and LP, and a proposed algorithm on the same dataset.

BR is a widely used baseline method for multi-label classification tasks. It treats each label as an independent binary classification problem and trains a separate binary classifier for each label. During the testing phase, each classifier predicts the presence or absence of its corresponding label. Then, the predictions are combined to generate the final set of labels for the input sample. Binary Relevance is a simple and efficient method, but it ignores the potential dependencies between labels and may not perform well when labels are highly correlated.

CC is a method that considers the label dependencies by creating a chain of binary classifiers. In this approach, the first classifier is trained on the input data, and each subsequent classifier is trained on the input data concatenated with the predictions of the previous classifiers. The order of the classifiers in the chain can be randomized or chosen based on some heuristic. Classifier Chains can capture the dependencies between labels and improve classification performance compared to Binary Relevance.

However, it requires more training time and is sensitive to the classifiers' order. Label Powerset is another method that considers the label dependencies by transforming the multi-label problem into a multiclass problem. In this approach, each unique combination of labels is treated as a separate class, and a classifier is trained to predict the class of each input sample. The Label Powerset method can handle many labels but can be computationally expensive due to the many possible label combinations. It may also suffer from the "curse of dimensionality" if the number of labels is too large.

The proposed algorithm is compared with BR, CC, and LP for multi-label classification, considering some potential factors for comparison:

1) **Performance** A proposed approach's performance will be compared with other algorithms as the most critical factor. The performance of different algorithms can be compared using evaluation metrics such as precision, recall, F1-score, accuracy, and others. The proposed approach may perform better or worse depending on the dataset and the characteristics of the problem.
2) **Scalability** The scalability of an algorithm is another factor to take into consideration. For example, there can be scalability issues with the LP algorithm, while the BR algorithm is generally more scalable when the number of labels is large.
3) **Interoperability** When the results need to be understood by humans, interpretability is an essential factor. Since CC model the dependency between labels, it may be easier to interpret than other algorithms.
4) **Complexity** In addition, complexity can be one of the factors to consider, affecting training times, memory requirements, and other aspects of the algorithm. As a result, there may be differences between the proposed approach and other algorithms in complexity.
5) **Data Distribution** Data attributes such as sparsity, imbalance, and noise can impact algorithm performance. Therefore, the proposed approach may have specific strengths or weaknesses based on the data distribution type.

Model performance for multi-label classification of dark web data with BR, CC, and LP algorithms:

Fig. 4, 5, and 6 illustrate the performance of the BR, CC, and LP algorithms in terms of accuracy and loss during both the training and testing phases. These figures provide valuable insights into the behavior of each algorithm over time, enabling a thorough analysis of their effectiveness and identifying potential areas for improvement in both training and testing scenarios.

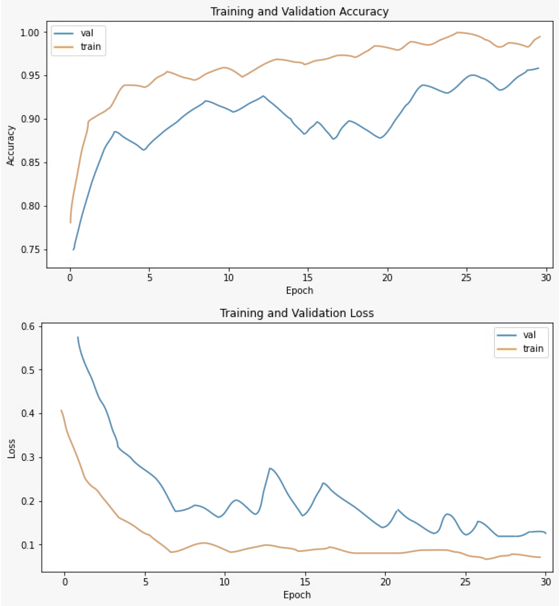

Fig. 4. Model accuracy and loss graphs for BR algorithm.

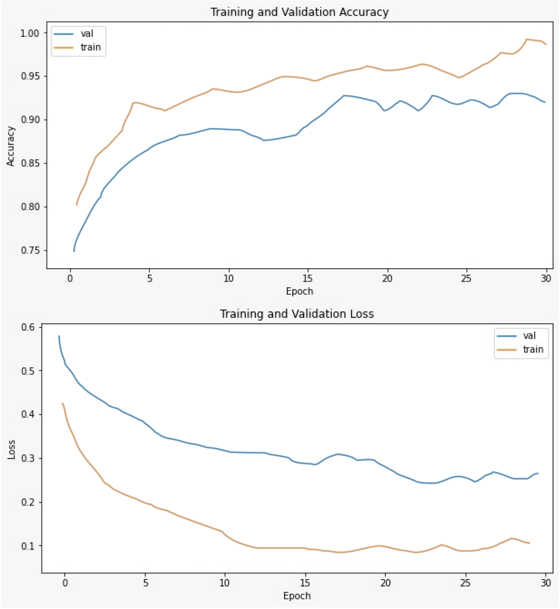

Fig. 5. Model accuracy and loss graphs for CC algorithm.

In the training and testing phases, accuracy and loss graphs are helpful indicators of the model's performance. High accuracy and low losses indicate that the model is learning and improving. Furthermore, when assessing the overall performance of a model, other metrics, such as precision, recall, and F1

TABLE I. COMPARISON OF PRECISION (PRC), RECALL (REC), AND F1 SCORE OF PROPOSED ALGORITHM (PRP ALG), BR, CC, AND LP

Algorithm	Prp Algo			BR			CC			LP		
Labels	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Business	0.70	0.78	0.73	0.60	071	0.71	0.72	0.75	0.74	0.68	0.66	0.60
Cybersecurity	0.40	0.67	0.50	0.47	0.65	0.59	0.48	070	0.56	0.39	0.62	0.48
Education	0.35	0.43	0.39	0.32	0.42	0.32	0.37	0.43	0.39	0.30	0.40	0.37
Entertainment	0.70	0.77	0.73	0.73	0.77	0.71	0.68	0.77	0.73	0.66	0.73	0.70
Finance	0.49	0.56	0.52	0.45	0.51	0.50	0.51	0.56	0.52	0.44	0.52	0.50
Food	0.93	0.95	0.94	0.91	0.98	0.96	0.91	0.95	0.94	0.91	0.96	0.91
Health	0.86	0.65	0.74	0.82	0.61	0.71	0.80	0.65	0.74	0.84	0.64	0.72
Literature	1.00	0.75	0.86	1.00	0.73	0.80	1.00	0.75	0.86	1.00	0.74	0.82
Nature	0.96	0.90	0.93	0.91	0.91	0.90	0.90	0.90	0.93	0.98	0.90	0.95
Network	0.86	0.86	0.86	0.89	0.85	0.80	0.79	0.86	0.86	0.89	0.89	0.89
Politics	0.29	0.67	0.40	0.20	0.66	0.41	0.18	0.67	0.40	0.24	0.69	0.43
Security	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Shopping	0.79	0.80	0.80	0.71	0.80	0.81	0.67	0.80	0.78	0.72	0.77	0.73

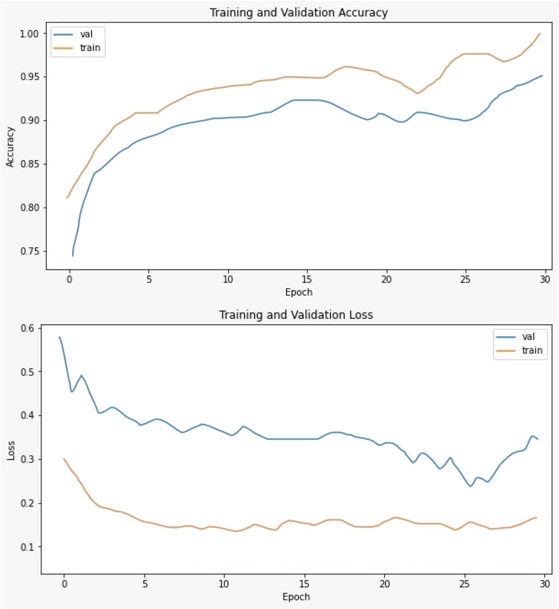

Fig. 6. Model accuracy and loss graphs for LP algorithm.

score, should also be considered.

C. Performance Evaluation of Algorithms

Comparison of precision, recall, and F1 scores provide insights into which algorithm performs better regarding identifying relevant labels. The F1 score is a harmonic mean of precision and recall. The accuracy of the predictions measures precision and recall by the ability to identify all relevant positive instances, while the precision of the forecasts measures recall.

Table I shows that comparing the proposed algorithm to the other three algorithms (BR, CC, and LP) for three labels (Business, Cybersecurity, and Education), the proposed algorithm has higher precision, recall, and F1 scores. Hence, the proposed algorithm outperforms the other algorithms for these labels. In addition, the proposed algorithm appears more accurate at identifying relevant instances within Entertainment than the other three algorithms.

In Finance, F1 scores for the baseline algorithm were the highest, while precision scores were the highest for the proposed algorithm. The proposed algorithm performs better than the other two algorithms in all but precision. Regarding Food, all algorithms performed well, with high precision, recall, and F1 scores. According to the proposed algorithm, the recall score was the highest, while the precision and F1 scores were the highest for the baseline and LP algorithms. BR algorithm had the highest recall and precision scores in Health, while the proposed algorithm had the highest precision and F1 scores.

For the Literature label, BR, LP, and the Proposed algorithm correctly identified labels for this category. For Nature, with F1 scores of 0.93 and 0.82, respectively, the Proposed Algorithm and LP perform well, while CC performs poorly with 0.80. In the Network category, the Proposed Algorithm, BR, and LP all achieve high F1 scores of 0.86, whereas CC achieves a slightly lower F1 score of 0.80. All algorithms perform reasonably well in this category.

In the politics category, the Proposed algorithm and BR scored 0.40, indicating they could not correctly identify the labels. On the other hand, there was a slight improvement in the F1 scores of LP and CC, respectively, with 0.43 and 0.41. In Security, a perfect score of 1.0 was achieved by all algorithms for precision, recall, and F1, indicating that all assigned labels were identified correctly. For the category of Shopping, BR, and LP have lower scores of 0.78 and 0.73, respectively, compared with the Proposed algorithm and CC.

Overall, the algorithms perform differently in different categories. However, most categories show that the Proposed algorithm and LP are better at identifying the labels for the given text data than BR and CC. The algorithm's poor performance in the Politics category demonstrates that category's characteristics affect the algorithm's effectiveness.

The number of labels per sample significantly affects multi-label classification using the proposed approach. Models may be unable to capture the complex relationships between features and labels when the number of labels is too low. This may cause the model to underperform on the test set and not generalize well. On the other hand, a model may suffer from the curse of dimensionality if there are too many labels per sample, which creates a very complex feature space. As a result, the model may struggle to learn effectively from the data, leading to poor performance. As a result, the optimal performance of the proposed approach depends on balancing the number of labels per sample.

V. Conclusion

The proposed work introduces a multi-label text classification approach to categorize deep and dark web content, aiming to predict multiple labels for a given text. Four machine learning algorithms were compared: the proposed algorithm, BR, CC, and LP. Evaluation metrics including precision, recall, and F1 score were used to assess their performance. The proposed algorithm exhibited significantly higher precision, recall, and F1 scores compared to the other three algorithms. Additionally, the study highlights the influence of the number of labels per sample on multi-label classification performance. Balancing the number of labels per sample is crucial to avoid poor results caused by either too few or too many labels per sample, which can lead to difficulties in capturing relationships between features and labels or the curse of dimensionality, respectively. In conclusion, this research proposes an efficient multi-label classification model for deep and dark web content analysis, demonstrating superior performance compared to existing methods, with potential applications in cybersecurity and law enforcement. Furthermore, the insights gained regarding the impact of the number of labels per sample can guide the development of future multi-label classification models. The multi-label text classification approach also enhances the model's capabilities to simultaneously learn entity recognition, relation extraction, and other related tasks, thus improving its overall performance.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[2] Y. He, D. Xin, V. Ganti, S. Rajaraman, and N. Shah, "Crawling deep web entity pages," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 355–364.

[3] Y. Li, G. Wu, and X. Wang, "Deep web data source classification based on text feature extension and extraction," *Infocommunications Journal*, vol. 11, no. 3, pp. 42–49, 2019.

[4] F. T. Ngo, C. Marcum, and S. Belshaw, "The dark web: What is it, how to access it, and why we need to study it," *Journal of Contemporary Criminal Justice*, p. 10439862231159774, 2023.

[5] S. Samtani, W. Li, V. Benjamin, and H. Chen, "Informing cyber threat intelligence through dark web situational awareness: The azsecure hacker assets portal," *Digital Threats: Research and Practice (DTRAP)*, vol. 2, no. 4, pp. 1–10, 2021.

[6] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. De Paz, "Classifying illegal activities on tor network based on web textual contents," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 35–43.

[7] V. Mahor, R. Rawat, A. Kumar, M. Chouhan, R. N. Shaw, and A. Ghosh, "Cyber warfare threat categorization on cps by dark web terrorist," in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2021, pp. 1–6.

[8] S. Jeziorowski, M. Ismail, and A. Siraj, "Towards image-based dark vendor profiling: An analysis of image metadata and image hashing in dark web marketplaces," in *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, 2020, pp. 15–22.

[9] G. Xu, Z. Wu, C. Li, J. Yan, J. Yuan, Z. Wang, and L. Wang, "Method of deep web collection for mobile application store based on category keyword searching," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 12th International Conference, SpaCCS 2019, Atlanta, GA, USA, July 14–17, 2019, Proceedings 12*. Springer, 2019, pp. 325–335.

[10] P. Mishra and A. Khurana, "Accuracy crawler: An accurate crawler for deep web data extraction," in *2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT)*. IEEE, 2018, pp. 25–29.

[11] A. Khare, A. Dalvi, and F. Kazi, "Smart crawler for harvesting deep web with multi-classification," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2020, pp. 1–5.

[12] S. Ghosh, A. Das, P. Porras, V. Yegneswaran, and A. Gehani, "Automated categorization of onion sites for analyzing the darkweb ecosystem," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1793–1802.

[13] L. Wang, A. Hawbani, and X. Wang, "Focused deep web entrance crawling by form feature classification," in *Big Data Computing and Communications: First International Conference, BigCom 2015, Taiyuan, China, August 1-3, 2015, Proceedings*. Springer, 2015, pp. 79–87.

[14] T. Sabbah and A. Selamat, "Modified frequency-based term weighting scheme for accurate dark web content classification," in *Information Retrieval Technology: 10th Asia Information Retrieval Societies Conference, AIRS 2014, Kuching, Malaysia, December 3-5, 2014. Proceedings 10*. Springer, 2014, pp. 184–196.

[15] H. M. Alghamdi and A. Selamat, "Topic detections in arabic dark websites using improved vector space model," in *2012 4th Conference on Data Mining and Optimization (DMO)*. IEEE, 2012, pp. 6–12.

[16] A. Dalvi, I. Siddavatam, A. Jain, S. Moradiya, F. Kazi, and S. Bhirud, "Element: Text extraction for the dark web," in *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2021*. Springer, 2022, pp. 537–551.

[17] O. Cherqi, G. Mezzour, M. Ghogho, and M. El Koutbi, "Analysis of hacking related trade in the darkweb," in *2018 IEEE international conference on intelligence and security informatics (ISI)*. IEEE, 2018, pp. 79–84.

[18] S. Nazah, S. Huda, J. H. Abawajy, and M. M. Hassan, "An unsupervised model for identifying and characterizing dark web forums," *IEEE Access*, vol. 9, pp. 112 871–112 892, 2021.

[19] H. Alnabulsi and R. Islam, "Identification of illegal forum activities inside the dark net," in *2018 international conference on machine learning and data engineering (iCMLDE)*. IEEE, 2018, pp. 22–29.

[20] C. Heistracher, F. Mignet, and S. Schlarb, "Machine learning techniques for the classification of product descriptions from darknet marketplaces." in *ICAI*, 2020, pp. 128–137.

[21] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[22] G. Nasierding and A. Z. Kouzani, "Comparative evaluation of multi-label classification methods," in *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2012, pp. 679–683.

[23] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde, "Binary relevance efficacy for multilabel classification," *Progress in Artificial Intelligence*, vol. 1, pp. 303–313, 2012.

[24] E. Montañes, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognition*, vol. 47, no. 3, pp. 1494–1508, 2014.

[25] A. Y. Taha and S. Tiun, "Binary relevance (br) method classifier of multi-label classification for arabic text." *Journal of Theoretical & Applied Information Technology*, vol. 84, no. 3, 2016.

[26] W. Liu and I. Tsang, “On the optimality of classifier chain for multi-label classification,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[27] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains: a review and perspectives,” *Journal of Artificial Intelligence Research*, vol. 70, pp. 683–718, 2021.

[28] J. C. Junior, E. Faria, J. Silva, and R. Cerri, “Label powerset for multi-label data streams classification with concept drift,” in *Proceedings of the 5th Symposium on Knowledge Discovery, Mining and Learning*. Faculdade de Computação-Universidade Federal de Uberlândia, 2017, pp. 97–104.

[29] W. Weng, Y.-W. Li, J.-H. Liu, S.-X. Wu, C.-L. Chen, W. Weng, Y. Li, J. Liu, S. Wu, and C. Chen, “Multi-label classification review and opportunities,” *J Netw Intell*, vol. 6, no. 2, pp. 255–275, 2021.

[30] X. Zhang, Q.-W. Zhang, Z. Yan, R. Liu, and Y. Cao, “Enhancing label correlation feedback in multi-label text classification via multi-task learning,” *arXiv preprint arXiv:2106.03103*, 2021.

[31] J. Ramos *et al.*, “Using tf-idf to determine word relevance in document queries,” in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1. Citeseer, 2003, pp. 29–48.

[32] Y. Zhou, “A review of text classification based on deep learning,” in *Proceedings of the 2020 3rd international conference on geoinformatics and data analysis*, 2020, pp. 132–136.