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

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Evolutionary Multiobjective Feature Selection for Sentiment Analysis

AYÇA DENIZ[1], (Member, IEEE), MERIH ANGIN[2], AND PELIN ANGIN[1], (Member, IEEE)

[1]Department of Computer Engineering, Middle East Technical University, Ankara, Turkey (e-mails: ayca.deniz@metu.edu.tr, pangin@ceng.metu.edu.tr)
[2]Department of International Relations, Koç University, İstanbul, Turkey (e-mail: mangin@ku.edu.tr)

Corresponding author: Merih Angin (e-mail: mangin@ku.edu.tr).

This research has been produced benefiting from the 2232 International Fellowship for Outstanding Researchers Program of TÜBİTAK (Project No: 118C309). However, the entire responsibility of the publication belongs to the owners of the research. The financial support received from TÜBİTAK does not mean that the content of the publication is approved in a scientific sense by TÜBİTAK.

ABSTRACT Sentiment analysis is one of the prominent research areas in data mining and knowledge discovery, which has proven to be an effective technique for monitoring public opinion. The big data era with a high volume of data generated by a variety of sources has provided enhanced opportunities for utilizing sentiment analysis in various domains. In order to take best advantage of the high volume of data for accurate sentiment analysis, it is essential to clean the data before the analysis, as irrelevant or redundant data will hinder extracting valuable information. In this paper, we propose a hybrid feature selection algorithm to improve the performance of sentiment analysis tasks. Our proposed sentiment analysis approach builds a binary classification model based on two feature selection techniques: an entropy-based metric and an evolutionary algorithm. We have performed comprehensive experiments in two different domains using a benchmark dataset, Stanford Sentiment Treebank, and a real-world dataset we have created based on World Health Organization (WHO) public speeches regarding COVID-19. The proposed feature selection model is shown to achieve significant performance improvements in both datasets, increasing classification accuracy for all utilized machine learning and text representation technique combinations. Moreover, it achieves over 70% reduction in feature size, which provides efficiency in computation time and space.

INDEX TERMS Binary classification, evolutionary computation, feature selection, multiobjective optimization, sentiment analysis.

I. INTRODUCTION

The significant advances in data storage, communication and processing technologies in recent years have given rise to the big data era, with a plethora of information flowing in from various data sources at high speeds. The high volume of data generated is useful to provide insightful information to decision-makers in various domains. Sentiment analysis, which provides automated extraction of opinions or feelings, is one of the techniques that play an essential role in decision-making processes [1]. It is also known as opinion mining, since it aims to extract subjective opinion from a piece of text [2]. Sentiment analysis has been gaining more attention recently, as it is a significant element of many real-world applications, including recommendation systems [3], analysis of product reviews [4], terrorist organization tracking [5], detection and analysis of critical events [6]–[8], real-time observation of public opinion [9], finance [10] and healthcare systems [11], [12]. Sentiment analysis can be defined as a polarity classification problem. This classification problem can be formed as a binary (positive vs negative) or multi-class (varying degrees of positive, negative and neutral) classification problem. Moreover, it can be applied at different levels, including analysis of words, sentences or whole documents. Recently, aspect-based sentiment analysis has also gained attention as a text may contain multiple aspects having different sentiments [13], [14].

At a high level, there exist three approaches to address the sentiment analysis task [15]: lexicon-based, machine learning-based, and hybrid approaches. Lexicon-based methods use a dictionary or corpus in which each word has a sentiment score [16]. This way, the sentiment of a sentence can be calculated using the sentiments of each word, combined using different techniques such as aggregation (e.g. majority voting). Although lexicon-based methods are easy to apply, they suffer from the lack of domain-specific dictionaries [17]. While machine learning techniques have achieved promising improvements over lexicon-based approaches, they require feature engineering for natural language processing (NLP) tasks [18]. More specifically, free-form textual data must be translated into a standard representation (vectorization) that the machine learning techniques can interpret. Hybrid approaches combine lexicon-based and machine learning-based methods for sentiment analysis.

Research on sentiment analysis is rapidly evolving as the number of new platforms, such as blogs and social media, where people continuously share their ideas have been on the rise. The abundance of such platforms has made large volumes of text data, including opinions and reviews, available for analysis of sentiments. Recent research has mainly focused on deep learning architectures for sentiment analysis tasks [19]–[24], as these architectures provide semantics information intrinsically through their hierarchical learning process [25]. On the other hand, deep learning requires a massive amount of training data to create accurate models.

Sentiment analysis faces challenges due to the existence of slang words, spelling mistakes [26] and ironic remarks in documents. One of the main challenges in sentiment classification is the high amount of data that contain irrelevant or redundant features [27], which adversely affect the performance of machine learning models [28]. Feature selection is one of the effective preprocessing techniques to eliminate features that have low or no contribution to the classification task [29]. There exist three main types of feature selection methods: filter-based, wrapper-based, and embedded [30]. Filter-based methods utilize metrics such as Chi-square to calculate the significance of a feature. On the contrary, wrapper-based methods utilize machine learning algorithms when deciding the most informative features. Wrappers generally perform better than filters [31], however they are more costly in terms of computation time and space. Finally, embedded methods perform feature selection while training the model, as they combine feature selection with the construction of the machine learning models.

Feature selection has been widely used for sentiment analysis in various domains and has proven to enhance the performance of sentiment classification [32], [33]. Previous studies mainly focused on filter [34] and wrapper [35] based feature selection methods. Although there exist feature selection methods that combine filter and wrapper based approaches for sentiment analysis [36], [37], all of them approach the problem in a single objective perspective. To the best of our knowledge, applying a multiobjective hybridized feature selection method to the sentiment analysis task has not been investigated yet.

In this paper, we propose a new hybrid multiobjective feature selection model for the sentiment analysis task, which harnesses the power of an entropy-based metric, i.e., Information Gain, and an evolutionary algorithm, i.e., Non-dominated Sorting Genetic Algorithm II (NSGA-II). Experiments with different machine learning and feature extraction techniques on the well-known Stanford Sentiment Treebank dataset demonstrate that our proposed model improves the learning performance of the sentiment analysis task considerably. Further, we introduce a new dataset: World Health Organization (WHO) Director-General's Speeches during part of the COVID-19 pandemic period (February - November 2020). This dataset consists of more than 10000 sentences labelled as positive, negative, or neutral. Replication of the experiments on the new dataset yields a similar outcome: our model significantly boosts the performance of the sentiment classification task.

The rest of this paper is organized as follows. In Section II, we provide related research about sentiment analysis, multiobjective feature selection, and feature selection methods applied for the sentiment analysis task. In Section III, we give the problem definition and describe the proposed model along with the utilized preprocessing, feature extraction and feature selection techniques. In Section IV, we share the experimental environment, including datasets and applied machine learning techniques. Then, we provide the experiment results in detail. Finally, we provide concluding remarks and future work directions in Section V.

II. RELATED WORK

Sentiment analysis has been a popular research topic due to its wide scope of applications, ranging from recommendation systems to finance [38]. Although sentiment analysis has been extensively studied in the literature, new studies continue to emerge as available data continually grow and become more complex. It is crucial to select the optimal feature subset for sentiment analysis [39] to achieve high performance. Therefore, feature selection is an indispensable preprocessing step, alleviating the burden caused by the high-dimensional data. Recently, Madasu and Elango [33] presented a detailed evaluation of different feature selection methods for sentiment analysis. They reported that feature selection methods, especially the ones that utilize ensemble techniques, obtain superior results by boosting the sentiment analysis performance. Ahmad et al. [40] reviewed feature selection methods used for sentiment analysis. They identified and presented the advantages and disadvantages of these methods. The authors suggested that metaheuristic algorithms perform well when selecting the optimal features for sentiment analysis. Shang et al. [41] presented a binary-based Particle Swarm Optimization (PSO) for feature selection in the sentiment analysis domain. Their algorithm was built to overcome the shortcomings of the traditional PSO algorithm, such as the update formula of velocity. Similarly, Kumar et al. [42] proposed a Firefly Algorithm for optimizing the feature sets to be used in sentiment analysis. They applied their algorithm to Hindi and English texts using SVM as the classifier. Gokalp et al. [43] proposed another wrapper-based feature selection method for sentiment analysis. The proposed model is based on a Greedy Algorithm that utilizes six different filter-based metrics, including Chi-square and ReliefF, in the construction of the model. Experiments on many public datasets showed that the model is more effective than conventional filter-based feature selection methods.

In the literature, there are three types of feature selection methodologies: filter-based, wrapper-based, and embedded. Filter-based methods utilize statistical information within the data. Some of the well-known metrics used by filter-based methods are Mutual Information, Information Gain, and Chi-square. Wrapper-based methods employ a search algorithm. Embedded methods combine the search process with classifier training. Wrapper-based feature selection methods generally perform better than filter-based methods [31]. Therefore, the recent literature in feature selection has mainly focused on wrapper-based methods. However, these methods are expensive in terms of computation time and space, as wrapper-based feature selection is an NP-hard problem [44]. Metaheuristic algorithms are known to be very efficient for NP-hard problems [45] and have been utilized by many researchers for feature selection in recent years. Al-Tashi et al. [46] presented a detailed review of multiobjective feature selection techniques and challenges. Kiziloz et al. [47] proposed three variants of multiobjective Teaching-Learning-Based Optimization algorithm for the feature selection task. Similarly, Sihwail et al. [48] proposed an improved version of Harris Hawk Optimization for the feature selection task. They presented three new search strategies to enhance the exploration capability of the hawks. Hu et al. [49] proposed a fuzzy cost-based Particle Swarm Optimization algorithm for multiobjective feature selection. Similarly, Zhang et al. [50] presented novel operators for the Artificial Bee Colony algorithm to tackle cost-sensitive multiobjective feature selection problems. Zhang et al. [51] employed differential evolution to improve the search operation of multiobjective feature selection tasks.

There exist studies that combine multiple feature selection methods to enhance the efficiency of the sentiment analysis task. Rasool et al. [17] proposed a hybrid feature selection method for sentiment classification. They selected promising features using different wrapper approaches and transferred them to the population of their Genetic Algorithm. Similarly, Ansari et al. [52] proposed another hybrid method for sentiment classification. They first applied two filter-based methods and extracted the most valuable features obtained by both methods. Then, they fed these features to two wrapper-based methods separately, namely, PSO and Recursive Feature Elimination, and reported that feature selection improves the classification performance tremendously. Pandey et al. [53] introduced another metaheuristic method, namely Cuckoo Search Algorithm, for sentiment analysis tasks. They utilized K-means to enhance the initialization process of their algorithm for faster convergence and better solution sets. Recently, Tubishat et al. [36] proposed an improved version of the Whale Optimization Algorithm (WOA) for sentiment analysis in Arabic texts. They combined Differential Evolution with Elite Opposition-Based Learning to boost the performance of WOA. Moreover, they utilized a filter-based feature selection method to feed valuable features to their algorithm. Hassonah et al. [37] introduced a hybrid feature selection method for sentiment analysis. Their method consists of a filter and wrapper-based approach. They analyzed the extracted features to find out which type of features (subjective, objective or emoticons) are more valuable in the sentiment analysis task.

III. FEATURE SELECTION MODEL

In this section, we formally describe the feature selection process for sentiment analysis, followed by the proposed evolutionary multiobjective feature selection model.

A. PROBLEM DEFINITION

Sentiment analysis can be considered as a polarity classification problem. The classification task is one of the fundamental problems in knowledge discovery. The accuracy of classification highly depends on the quality of the data. Therefore, it is vital to preprocess the data to extract valuable information. Especially in real-world applications, the data amount is generally high, and there exist many redundant or irrelevant features that have no contribution to the classification task.

Feature selection is an important preprocessing step for classification. It aims to find the most informative features that can represent the data. Through feature selection, the training time of the model is also reduced. Moreover, the learning performance of the model improves as unnecessary features will not clutter the model. However, the feature selection task can be challenging, as it is a combinatorial optimization problem.

Feature selection requires optimizing two objectives, minimizing the number of features and maximizing the classification performance. This optimization task can be formally defined as follows:

$$\begin{aligned} & min\ obj_1 \\ & max\ obj_2 \\ & subject\ to \\ & \qquad obj_1 = |d| \\ & \qquad obj_2 = performance(d) \\ & where\ d \subseteq D \end{aligned} \tag{1}$$

where D is the data with all features, and d is the selected feature subset of D. In this equation, obj_1 and obj_2 indicate the first and second objectives, respectively. Regarding these objectives, we aim to reduce the number of features, i.e., obj_1, while we try to improve the classification performance, i.e., obj_2. In this study, we utilize accuracy as the performance metric. Accuracy is the ratio of the number of correctly classified instances over the number of all instances. According to the feature selection definition, an ideal solution would have a 100% classification accuracy using only one feature.

In a multiobjective optimization problem, there might be a solution set instead of only one solution. The reason is that, one solution might be good at achieving one objective, while another solution is good at achieving another. To illustrate, in

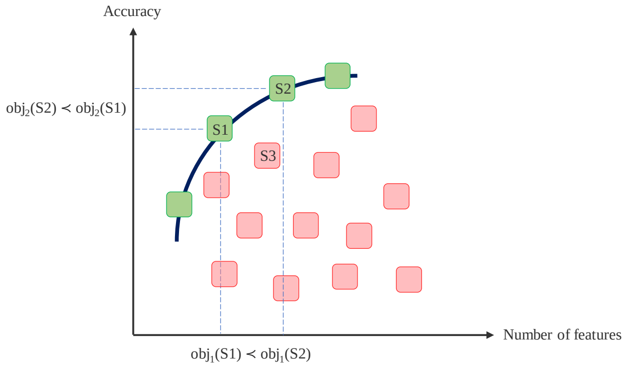


FIGURE 1: Sample solutions fitting to a Pareto curve for the two objectives of the multiobjective optimization problem.

Figure 1, we provide sample solutions for a feature selection task in which the two objectives defined above are optimized. In this figure, the solutions in green fit on a Pareto curve. These solutions are called non-dominated solutions, as they are not dominated by any other solution in both objectives. On the other hand, the red-colored solutions are dominated in both objectives by at least one other solution. For example, solution S1 is better than solution S3 in both objectives as it has fewer features and higher accuracy, as given by the inequalities below:

$$\begin{aligned} obj_1(S1) &< obj_1(S3) \\ obj_2(S1) &> obj_2(S3) \end{aligned} \tag{2}$$

As a result, S1 dominates S3, as represented below:

$$S1 \prec S3 \tag{3}$$

With a similar comparison, it can be seen that solution S1 cannot dominate solution S2. The number of features in S1 is less than the number of features in S2, but the accuracy of S2 is higher than the accuracy of S1. Hence, they are non-dominated solutions as they have better results in different objectives. As a result, these non-dominated solutions are presented as the final solution set for the problem.

B. PROPOSED MODEL

The flowchart of the proposed feature selection model is depicted in Figure 2. The algorithm begins by applying preprocessing to the raw data. After preprocessing is completed, features are extracted. As soon as the features are ready, the feature selection process begins. Feature selection in our model comprises two parts: filter and wrapper-based. With this process, the most promising features for the sentiment classification task are extracted. All the mentioned steps are explained in detail in the subsections below.

1) Preprocessing

Preprocessing is a crucial phase that affects the performance of classifiers [54]. With this step, the redundant data in the raw dataset are filtered out, as they do not have a meaningful contribution to the classification task. Moreover, reducing the dimensionality of the data speeds up the training process. We utilized the NLTK[1] library for preprocessing operations. In our proposed model, the preprocessing phase is four-fold:

a: Conversion to lowercase

In this step, all the words in all sentences are converted to lowercase. Without this operation, the model treats a word with a capital letter different from the same word without any capital letters, which could increase data sparsity and decrease the prediction accuracy of the model.

b: Punctuation removal

In this step, all punctuation marks are removed from the sentences. Similar to the previous step, the aim is to lower data sparsity, as the model cannot discriminate between punctuation and other characters.

c: Tokenization

In this step, all individual words are identified and split from each other. With tokenization, the sentences are split into minimal meaningful units which are later used in feature extraction.

d: Stop words removal

Stop words are the words that occur in texts with high frequencies but do not add a specific meaning to the text, such as *a*, *an*, *the*, *of*, etc. Therefore, in this step, stop words are removed so that only significant words are left for the training part.

2) Feature Extraction

There exist many feature extraction techniques to translate free-form textual data into a standard representation that machine learning techniques can interpret. In order to show that our model is viable regardless of the feature extraction technique, we tested it with different techniques separately. In this work, we utilized two feature representation techniques, *Bag-of-Words* and *GloVe*, which have different strengths and weaknesses.

a: Bag-of-Words

Bag-of-Words (BoW) is one of the basic and well-known text representation techniques [55]. BoW converts arbitrary texts into fixed-length vectors. In BoW, each sentence is represented as a vector s = $<x_1, x_2, \ldots, x_n>$ where x_i denotes the number of occurrences of the i-th token and n is the total number of unique tokens in all sentences. Therefore, the BoW method does not consider word orders when generating the features. Hence, the syntactic and semantic relationships are lost in this method. For example, assuming there are two sentences in the dataset: (i) 'I love tea, but I hate coffee', and (ii) 'I love coffee, but I hate tea'. The unique tokens

[1] https://www.nltk.org

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3118961, IEEE Access

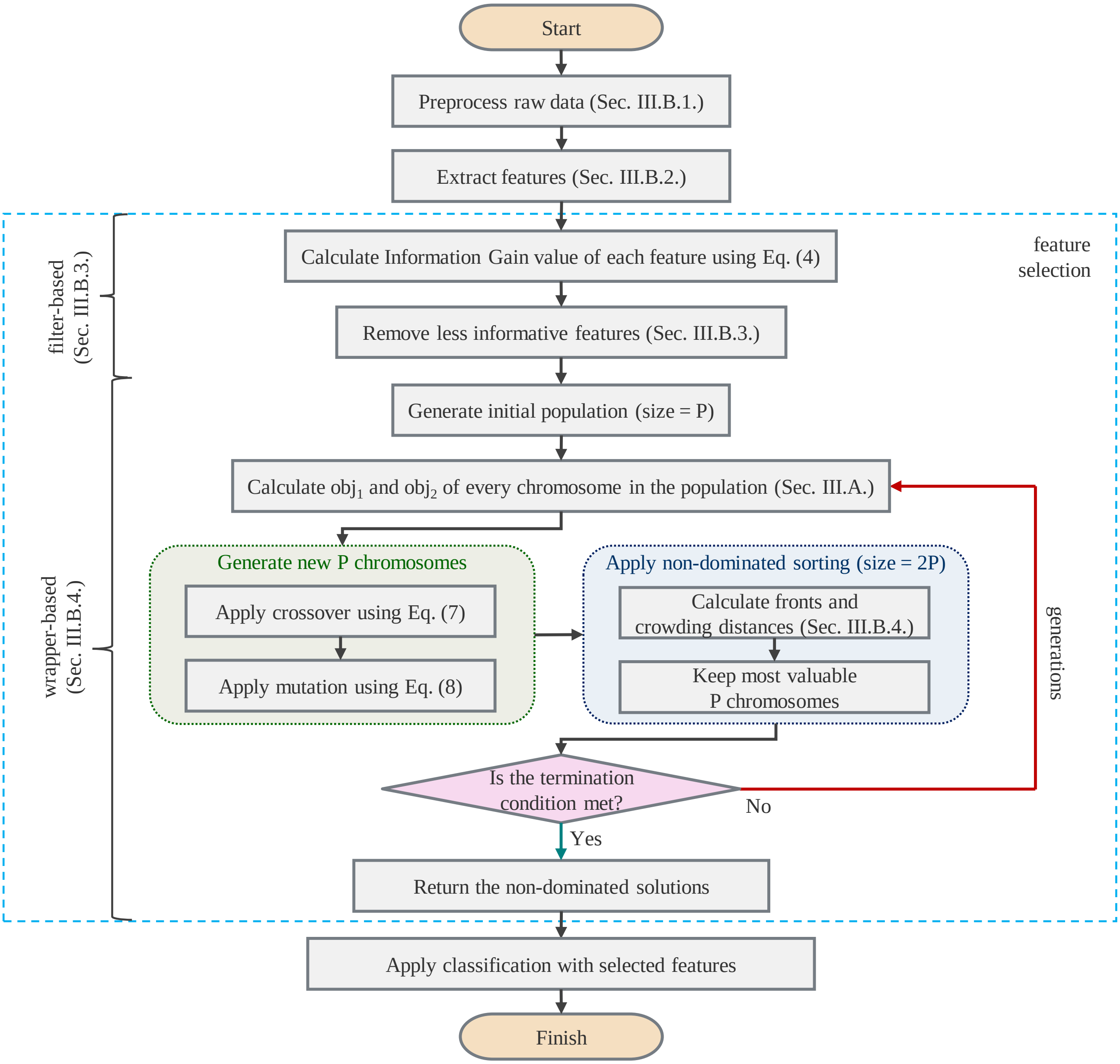

Start
Preprocess raw data (Sec. III.B.1.)
Extract features (Sec. III.B.2.)
feature selection
filter-based (Sec. III.B.3.)
Calculate Information Gain value of each feature using Eq. (4)
Remove less informative features (Sec. III.B.3.)
wrapper-based (Sec. III.B.4.)
Generate initial population (size = P)
Calculate obj$_1$ and obj$_2$ of every chromosome in the population (Sec. III.A.)
Generate new P chromosomes
Apply crossover using Eq. (7)
Apply mutation using Eq. (8)
Apply non-dominated sorting (size = 2P)
Calculate fronts and crowding distances (Sec. III.B.4.)
Keep most valuable P chromosomes
generations
Is the termination condition met?
No
Yes
Return the non-dominated solutions
Apply classification with selected features
Finish

FIGURE 2: The proposed feature selection model.

(features) for this dataset will be {'I', 'love', 'tea', 'but', 'hate', 'coffee'}. Although the two sentences have different meanings, their vector representations with BoW will be the same: <2, 1, 1, 1, 1, 1>.

In this study, every unique word in the dataset represents a feature. In our BoW representation, we construct a vector for every sentence in the dataset.

b: GloVe

GloVe is one of the well-known and effective pre-trained word embeddings [56]. A word embedding can simply be described as representing each word of a document with a real-valued feature vector, where words with similar meanings have a similar representation. The feature vectors are calculated via training a neural network using a large number of documents. This training process utilizes word positions in the documents. As a result, it is possible to capture semantic relations with word embeddings [24]. The famous example that demonstrates the existence of semantic relations is as follows: Having the feature vectors of the words *King*, *Queen*, *man*, and *woman*, if we subtract the vector for *man* from the vector for *King*, and add the vector for *woman* to it, the result becomes the feature vector of the word *Queen*. This example shows that the model automatically learns the male/female relationship. One problem with word embeddings is that they may not consider the context [23]. For example, the words *beetle* as a car and *beetle* as an animal are represented with the same vector in GloVe.

In this study, we employed 50-dimensional GloVe vectors[2]. Each dimension of these vectors represents a feature. We concatenated the GloVe word vectors to construct the vector representation of a sentence. For example, the words 'I', 'love', and 'tea' have the following vectors in GloVe: <0.118, 0.152, ..., 0.921>, <-0.138, 1.140, ..., 0.289>, and <-0.449, -0.002, ..., -0.902>, respectively. Consequently, the vector representation of 'I love tea' will be as follows: <0.118, 0.152, ..., 0.921, -0.138, 1.140, ..., 0.289, -0.449, -0.002, ..., -0.902>. We set the maximum length for a sentence as the upper quartile value of the number of tokens in all sentences. We padded the vector with zeros when the word count in the sentence was smaller than the specified length.

3) Filter-based feature selection

In the filter-based feature selection part of our model, we utilize the Information Gain metric [57]. Information Gain measures the information amount that a single feature carries in a set of features. Information Gain of a feature F is calculated with the following formula:

$$IG(D,F) = Entropy(D) - \sum_{u \in U} \frac{|D_u|}{|D|} Entropy(D_u) \quad (4)$$

where D is the data with all features and instances, F is the particular feature, U is the set of all the unique values for the related feature, and D_u is a subset of D, having the instances in which the value of F is u. $|D|$ and $|D_u|$ are the number of instances in D and D_u, respectively. The entropy of a subset S of the data is calculated as follows:

$$Entropy(S) = -\sum_{c \in C} p_c \log_2 p_c \quad (5)$$

where C is the set of all classes in the dataset and p_c is the ratio of the number of instances in the c-th class over the number of all instances in S.

In the literature, it is common to filter out the words that occur only once as they do not provide any predictive power [58]. By building on this idea, we filter out the words whose Information Gain value is below a certain threshold. However, it is not easy to choose a generic threshold value that would work well for all datasets. For this reason, we leverage information conveyed by the dataset itself to determine the threshold value. Consequently, in our model, we first calculate the Information Gain value of each feature in the dataset. Then, we compute the median value and set it as the threshold. Finally, we filter out the features whose values are less than the threshold as their predictive power is low. We call this procedure Information Gain Filtering (IGF). Choosing a smaller threshold value (e.g. first quartile value) would lead to the elimination of discriminative features for sentiment analysis. On the other hand, selecting this value larger (e.g. third quartile value) would prevent most features with low predictive power from being filtered out, which would worsen the learning performance.

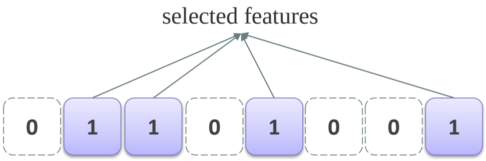


FIGURE 3: A sample chromosome.

4) Wrapper-based feature selection

In the wrapper-based feature selection part of our model, we apply the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [59]. NSGA-II is a well-known and efficient multiobjective optimization algorithm. With regard to the evolutionary nature of this algorithm, every possible solution is represented with a chromosome/individual I as below.

$$I = [f_1, f_2, ..., f_N] \quad (6)$$

where N is the total number of features in the dataset and f_i is the ith feature in the dataset. A sample chromosome is also depicted in Figure 3. Each chromosome's length is the total number of features in the dataset. The value of each segment can be either 1 or 0, indicating that a feature is selected or not, respectively, as given below.

$$f_i = \{0,1\} \qquad \text{for } i = 1...N \quad (7)$$

In the figure, the features two, three, five, and eight are selected. Accordingly, the first objective (number of features) for this chromosome becomes four. In order to calculate the second objective (accuracy), the remaining features (one, four, six, and seven) are filtered out, and only the selected features are used to train a classifier.

The NSGA-II algorithm in our study executes as follows. First, an initial population that consists of randomly generated chromosomes is generated. Then, the values of both objectives are calculated for every individual in the population. With the determination of the population, the first generation begins. Similar to a standard genetic algorithm, crossover and mutation operators are applied to randomly selected individuals (parents) to create new individuals (children) as many as the population size. With crossover and mutation operators, we aim to increase the diversity in the population.

We utilized the half-uniform crossover operator in our study. Let C_1 and C_2 be two chromosomes in the population. Two new chromosomes, C_3 and C_4, are generated using the crossover operation between C_1 and C_2, respectively. The equation below depicts the generation of C_3:

$$C_{3i} = \begin{cases} C_{1i}, & \text{if } C_{1i} = C_{2i} \\ rand(0,1), & \text{otherwise} \end{cases} \qquad \forall i \in C_1 \quad (8)$$

[2] https://nlp.stanford.edu/projects/glove/

where C_3 is the new chromosome and C_{1i}, C_{2i}, and C_{3i} are the i-th features in the chromosomes C_1, C_2, and C_3, respectively. C_4 is generated over C_2 in a similar fashion.

For mutating the newly generated chromosomes, we utilize the bit-flip mutation operator. Bit-flip mutation alters the chromosome as given in the equation below:

$$C'_i = \{1 - C_i : P(i) \geq MP\} \qquad \forall i \in C \tag{9}$$

where C' is the mutated chromosome, C'_i and C_i are the i-th features in the chromosomes C' and C, $P(i)$ is the randomly generated probability that the feature i is mutated, and MP is the predefined mutation probability which is shared in Section IV-A3.

After crossover and mutation operations are applied in the population, all new individuals are evaluated in terms of both objectives. Particularly, NSGA-II is an elitist algorithm. Therefore, the new individuals do not necessarily replace the existing individuals, but rather all individuals are combined in a pool, doubling the population size. To continue its execution, NSGA-II selects the better half of the pool as the next generation. However, due to having two objective values, selecting the better half is not a straightforward process. For this purpose, we use the non-dominated sorting algorithm, a methodology to compare the individuals in a multiobjective environment.

The non-dominated sorting algorithm divides the individuals into multiple fronts, as many fronts as required according to the dominance relationship. All the individuals that are not dominated by any other individual constitute the first front. Similarly, all the individuals that are dominated only by the individuals in the first front, but not dominated by any other individuals constitute the second front. This operation is repeated until all the individuals are assigned into a front. In comparison, any individual assigned to a front with a smaller front number is better than any individual that is assigned to a front with a larger front number.

Crowding distance is used to compare the individuals within the same front. The crowding distance values of the individuals are determined considering their neighbors. The half perimeter of the rectangle including the nearest left and right neighbor individuals in the same front denotes the crowding distance of the related individual. The crowding distance value of an individual (solution), S, is calculated as follows:

$$CD(S) = \sum_{o \in O} \frac{|S_{o+1} - S_{o-1}|}{|f_o^{max} - f_o^{min}|} \tag{10}$$

where O is the set of all objectives, S_{o+1} and S_{o-1} are the o-th objective values of the immediate neighbors of S, and f_o^{max} and f_o^{min} are the maximum and minimum values obtained for the o-th objective. The two extreme individuals, one individual having the maximum accuracy value and the one individual having the minimum number of features, are provided with the maximum crowding distance values for the specific front. Once all the individuals are assigned a crowding distance value, the individual having a higher crowding distance is considered better. Application of the non-dominated sorting algorithm for determination of the better half as the next population, concludes the generation. The algorithm iterates for a predetermined number of generations, and finally reports the non-dominated solutions of the final population as the result.

For clarity, we also provide the algorithm of our proposed model in Algorithm 1.

Algorithm 1: Algorithm of the proposed model.

```
instances: input data
FE: feature extraction technique
ML: machine learning technique

// apply preprocessing
instances ← ConvertToLowercase(instances);
instances ← RemovePunctuation(instances);
instances ← Tokenize(instances);
instances ← RemoveStopWords(instances);

// extract features
features ← ExtractFeatures(instances, FE);

// apply filter-based feature selection
ig_values ← CalculateInformationGain(features);  // Eq. 4
threshold ← Median(ig_values);
features ← InformationGainFiltering(features, threshold);

// apply wrapper-based feature selection
population ← GeneratePopulation(features);
population ← CalculateFitnessValues(population, ML);
for (g ← 1 to number_of_generations) do
    for (p ← 1 to population_size) do
        $parent_1$, $parent_2$ ← SelectParents(population);
        child ← Crossover($parent_1$, $parent_2$);  // Eq. 8
        child ← Mutation(child);  // Eq. 9
        population ← population ∪ child;

    // population size is doubled, keep better half
    fronts ← NonDominatedSort(population);
    fronts ← CalculateCrowdingDistance(fronts);  // Eq. 10
    population ← KeepBetterHalf(fronts);

print ($fronts_1$); // most valuable feature subsets

Function NonDominatedSort($P$):
    $i$ = 1;
    while $P \neq \varnothing$ do
        $F_i$ = $\varnothing$;
        foreach $p \in P$ do
            $n$ = 0;
            foreach $q \in P$ do
                if $q \prec p$ then
                    $n = n + 1$;
            if $n$ = 0 then
                $F_i = F_i \cup$ {p};
        $P = P \setminus F_i$;
        $i = i + 1$;
    return $F$; // $F$ consisting of all fronts {$F_1$, $F_2$, ...}
```

TABLE 1: Sample instances from the SST dataset.

index	sentence	train (1), test (2) or validation (3)	sentiment score
1601	If you enjoy more thoughtful comedies with interesting conflicted characters; this one is for you.	3	0.91667
5050	So original in its base concept that you can not help but get caught up.	1	0.88889
4263	I have two words to say about Reign of Fire.	1	0.5
8217	Scene-by-scene, things happen, but you'd be hard-pressed to say what or why.	2	0.31944
5800	The most offensive thing about the movie is that Hollywood expects people to pay to see it.	1	0.18056
6217	Plodding, poorly written, murky and weakly acted, the picture feels as if everyone making it lost their movie mojo.	1	0

TABLE 2: Statistics of the sentences in the SST dataset.

description	before preprocessing	after preprocessing
total count of sentences	11027	11027
total count of unique tokens in all sentences	16987	18296
average number of tokens in all sentences	16.1	9.3
standard deviation of the number of tokens in all sentences	8.2	4.7
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	10	6
50% percentile (median) of the number of tokens in all sentences	15	9
75% percentile (upper quartile) of the number of tokens in all sentences	22	12
maximum number of tokens in all sentences	50	28

TABLE 3: Number of instances for each sentiment class in the SST dataset.

sentiment	train set size	test set size
score $>$ 0.5 (positive)	4096	1033
score $=$ 0.5 (neutral)	197	43
score $<$ 0.5 (negative)	3824	1049

IV. EXPERIMENTS

In this section, we first describe the experimental setup, including utilized datasets, machine learning techniques, and parameter settings. Then, we present and discuss the experiment results.

A. EXPERIMENTAL SETUP

We carried out the experiments on a computer with Intel Core i7-9700K Eight-Core Processor with a 3.6 GHz clock rate and 16 GB of main memory. We used Python for implementation.

1) Datasets

We evaluated the performance of our model on two datasets. The first dataset is Stanford Sentiment Treebank (SST), which is one of the well-known datasets widely used in sentiment analysis studies in the literature [21], [23], [24]. The second dataset consists of the speeches of the World Health Organization Director-General in the pandemic period. These two datasets are briefly described below.

a: Stanford Sentiment Treebank

The Stanford Sentiment Treebank (SST) was introduced in 2013 by Socher et al. [60]. The dataset contains labelled training and test sets. In the dataset, there exist more than 10,000 sentences with more than 200,000 phrases obtained from movie reviews. Sample instances from SST dataset are provided in Table 1. Moreover, we report statistics of the sentences in the dataset in Table 2. Furthermore, in Table 3, we share the total number of instances for each sentiment in training and test sets separately. In the experiments, we filtered out the neutral-labelled instances as our study is on binary classification.

b: WHO Director-General's Speeches

WHO announced the COVID-19 disease as a pandemic in March 2020. Since then, the virus has rapidly spread all around the world. As of September 3, 2021, more than 4.5 million deaths and around 219 million cases have been recorded globally [61].

For this study, we collected the WHO Director-General's speeches during the pandemic period (between February 2020 and November 2020). Then we asked four annotators to label the sentences in these speeches in three categories: positive, neutral and negative. Sample instances from the WHO Speeches dataset are provided in Table 4. Moreover, we report statistics of the sentences in the dataset in Table 5. In Table 6, we share the total number of instances for each sentiment category. In the experiments, we filtered out the neutral-labelled instances as our study is on binary classification and we applied 5-fold cross-validation on the dataset to prevent bias.

2) Applied Machine Learning Techniques

There exist many effective machine learning techniques for the classification task. We evaluated the performance of our model using two machine learning techniques which are

TABLE 4: Sample instances from the WHO dataset.

sentence	sentiment
This marked one of the greatest public health achievements of all time.	positive
That is when you can clearly see what works, what doesn't and what you need to improve.	neutral
However, the COVID-19 pandemic hurt momentum as polio and immunization efforts were suspended.	negative

TABLE 5: Statistics of the sentences in the WHO dataset.

description	before preprocessing	after preprocessing
total count of sentences	7357	7357
total count of unique tokens in all sentences	6801	7028
average number of tokens in all sentences	18.7	10.1
standard deviation of the number of tokens in all sentences	9.3	5.4
minimum number of tokens in all sentences	1	0
25% percentile (lower quartile) of the number of tokens in all sentences	12	6
50% percentile (median) of the number of tokens in all sentences	18	10
75% percentile (upper quartile) of the number of tokens in all sentences	24	13
maximum number of tokens in all sentences	70	57

TABLE 6: Number of instances for each sentiment class in the WHO dataset.

sentiment	instance size for 5-fold CV
positive	5355
neutral	2718
negative	2002

briefly described below. We utilized the scikit-learn[3] implementation of these techniques.

a: Logistic Regression

Logistic Regression (LR) builds a probabilistic classification model. It is known as an easy-to-use and efficient classifier [62]. It estimates an item's class by applying the Sigmoid function, which is given below:

$$P(Y = 1 \mid X, \theta) = \frac{1}{1 + e^{-\theta X}} \tag{11}$$

where X is the input data, θ is the coefficient values for the input, and Y is the probability of an item belonging to class 1.

b: Support Vector Machines

Support Vector Machines (SVM) builds a linear classification model [63]. SVM maps data points into space to find the best hyperplane that separates the classes. It aims to maximize the distance between the support vectors (closest data points to the hyperplane) and the hyperplane with regard to the equation below:

$$\begin{aligned} &minimize\ ||w||\ in\ (w, b) \\ &subject\ to \\ &\qquad y_i(w^T x_i + b) \geq 1 \quad for\ i = 1...N \end{aligned} \tag{12}$$

where w, b, x, y are the weight, bias, input and output vectors respectively, and N is the number of instances.

TABLE 7: Parameter settings of all algorithms and techniques.

alg./tech.	parameter	value
NSGA-II	population size	100
	number of generations	200
	mutation ratio	2%
	crossover ratio	100%
IGF	threshold	median value
GloVe	max. token count for each sentence	upper quartile value
LR	solver	lbfgs
	multi_class	ovr
	max_iter	1000
SVM	C	0.1

3) Parameter settings

Table 7 presents the parameter settings of all the algorithms and techniques used in our study. Deniz et al. [64] report that the NSGA-II algorithm achieves better results as the population size and number of generations grow larger. Furthermore, they suggest that an increase in population size negatively affects the computation time more than an increase in the number of generations. Therefore, in this study, we selected the *population size* as *100* and the *number of generations* as *200*. As the NSGA-II algorithm is elitist in its nature, it keeps a copy of the parents in the pool of individuals for the next generation. Therefore, we set the *crossover ratio* as *100%* to increase the diversity inside the population. Moreover, we set the *mutation ratio* as *2%* to increase the exploration space of the algorithm.

For IGF, we set the *threshold* value as the *median* of information gain values of the features. All features having an information gain value less than the median are filtered out, as they have less predictive power. When using GloVe as the feature extraction technique, we represented each sentence with the same vector size. Therefore, the sentences having fewer tokens than the threshold value are padded with empty vectors, and the sentences having more tokens are cut off

[3] https://www.scikit-learn.org

TABLE 8: Comparison results of the proposed algorithm with other methods in terms of accuracy and number of features.

(a) The SST dataset.

model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	accuracy	# of features	accuracy	# of features	accuracy	# of features	accuracy
baseline	18296	0.761	18296	0.755	600	0.693	600	0.688
IGF	9434	0.803	9434	0.803	300	0.677	300	0.673
NSGA-II	8446	0.798	8327	0.794	162	0.732	194	0.722
IGF + NSGA-II	3972	0.845	4135	0.836	83	0.717	93	0.712

(b) The WHO dataset.

model	BoW				GloVe			
	LR		SVM		LR		SVM	
	# of features	accuracy	# of features	accuracy	# of features	accuracy	# of features	accuracy
baseline	7028	0.840	7028	0.840	650	0.798	650	0.800
IGF	4038	0.850	4038	0.851	325	0.800	325	0.803
NSGA-II	2879	0.853	3228	0.857	172	0.829	182	0.832
IGF + NSGA-II	1704	0.864	1728	0.861	91	0.825	95	0.828

from the threshold value. We set the threshold, i.e., the *maximum token count for each sentence*, as the *upper quartile value* of the number of tokens in all sentences. For LR, we set the *solver* parameter as *lbfgs* and *multi_class* parameter as *ovr*, since we apply it on a binary classification problem. Finally, we set the maximum number of iterations (*max_iter*) taken by the solver to converge as *1000*. For SVM, the regularization parameter, i.e., C, is an important parameter for performance. When it increases, training error decreases, whereas computation time massively increases as it tries to find a smaller-margin hyperplane that separates the classes. Therefore, we set C as *0.1* in our implementation.

B. EXPERIMENT RESULTS

In this section, we report the experimental results. Table 8 presents the accuracy and number of features achieved by various algorithms combined with feature extraction and machine learning techniques in both datasets. Baseline results (preprocessed data) are given in the first row. In the second row, the results when only IGF is applied (preprocessed data + IGF) are shared. In the next row, the results when only NSGA-II is applied (preprocessed data + NSGA-II) are given. The results for the combined model (preprocessed data + IGF + NSGA-II) are presented in the last row of the table. It can be clearly seen that the proposed model achieves a significant increase in accuracy with much fewer features as compared to the baseline.

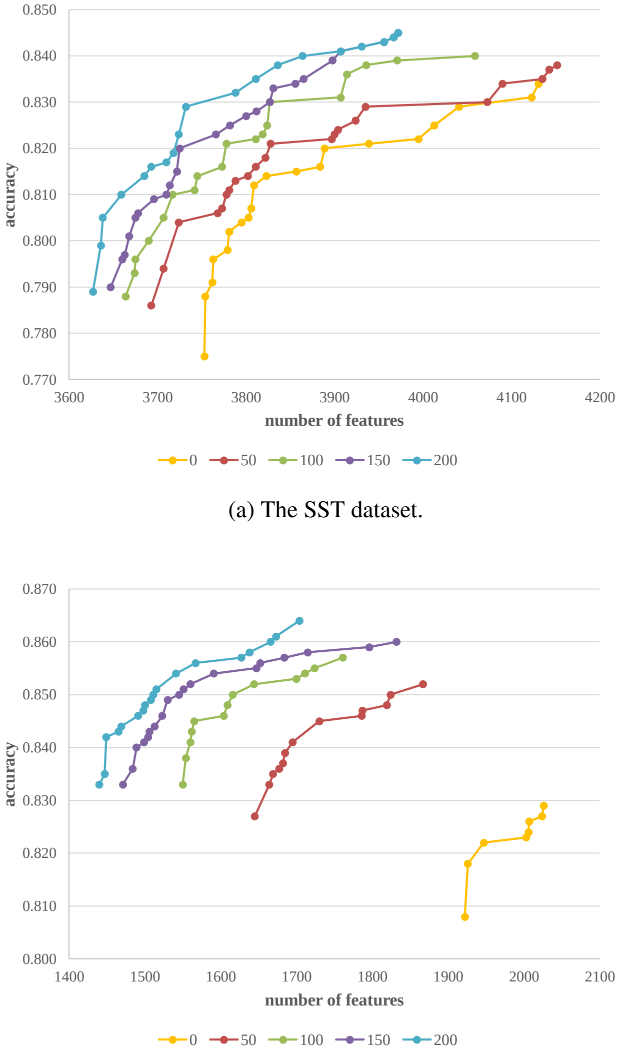
When we compare feature extraction techniques, BoW achieves higher accuracy values than GloVe. In terms of decreasing the number of features, both techniques manage to achieve a reduction of around 70%. We note that the results of GloVe might improve if a longer representation is chosen rather than the 50-dimensional GloVe vectors. Nevertheless, we can clearly see an improvement in accuracy over the baseline with our proposed model even for this version of GloVe.

When we compare machine learning techniques, LR achieves higher accuracy values and lower number of features than SVM. However, SVM runs faster than LR. For example, in baseline results for SST, the computation time of SVM is 0.8 seconds, whereas the computation time of LR is 8.1 seconds. After our proposed model decides the most valuable features, their execution times become 0.3 seconds and 1.3 seconds for SVM and LR, respectively.

In Figure 4, we present the non-dominated solutions obtained through the generations on a two-dimensional plot. In the subfigures, the number of features and accuracy values are given in the x- and y-axis, respectively. We report the results up to 200 generations, in intervals of 50. Significant improvements in terms of both the number of features and accuracy are observed as the number of generations increases. For example, initially, the number of features is about 2000 and accuracy is about 82% for the WHO dataset. With the proposed model, the number of features goes down to about 1450, and accuracy goes up to about 86%.

We provide the initial and final populations in Figure 5 to show that the proposed model evolves to approximate the optimal solution. The figures show that the initial population improves throughout the generations and gets closer to the ideal point, i.e., the point where the number of features is one and accuracy is 1.00. The individuals in the initial population are more scattered. In contrast, the non-dominated solutions in the final population fit to a Pareto-like curve as suggested in the Problem Definition (see Section III-A).

In Figure 6, we share the improvements in terms of the number of features, accuracy and execution time after the proposed algorithm is applied with the LR classifier. The percentages above the bars in the subfigures present the



(a) The SST dataset.

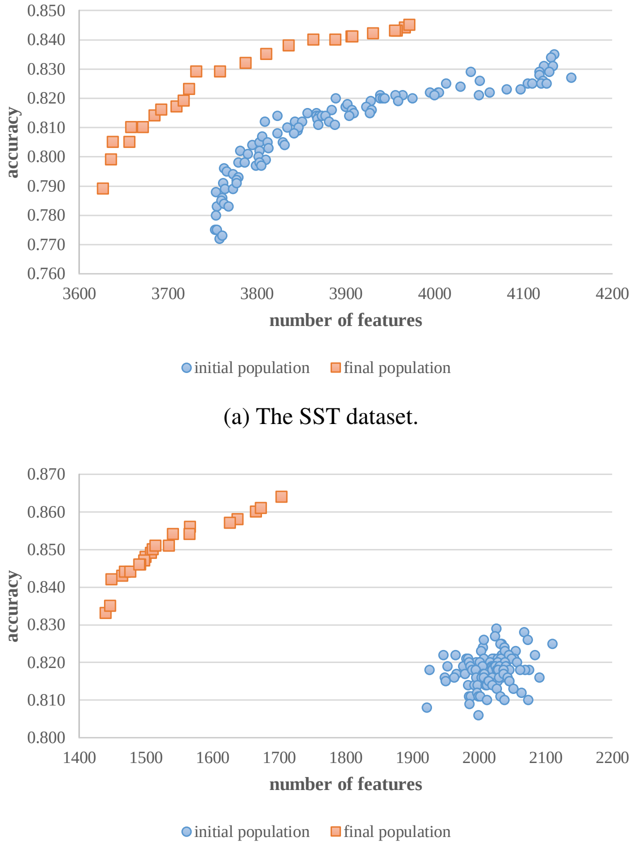
(b) The WHO dataset.

FIGURE 4: The evolution of the non-dominated solutions through generations.

amount of improvement in the related category and dataset. The figures show that the proposed algorithm decreases the number of features in the SST dataset by 78%. As the amount of data decreases, computation time reduces as well. We observe an 84% gain in the execution time of the classifier. Moreover, the proposed algorithm boosts accuracy by around 8%. Similar improvements are observed for the other dataset in the figure.

Upon the above findings, a chi-square test of independence was performed to examine the relation between the results of the baseline and the proposed model. The relation between these variables was significant, χ^2 (1, N = 2082) = 47.0388, $p < 0.001$. The proposed model significantly improves the performance of the sentiment classification task.

In order to verify the effectiveness of the proposed model,



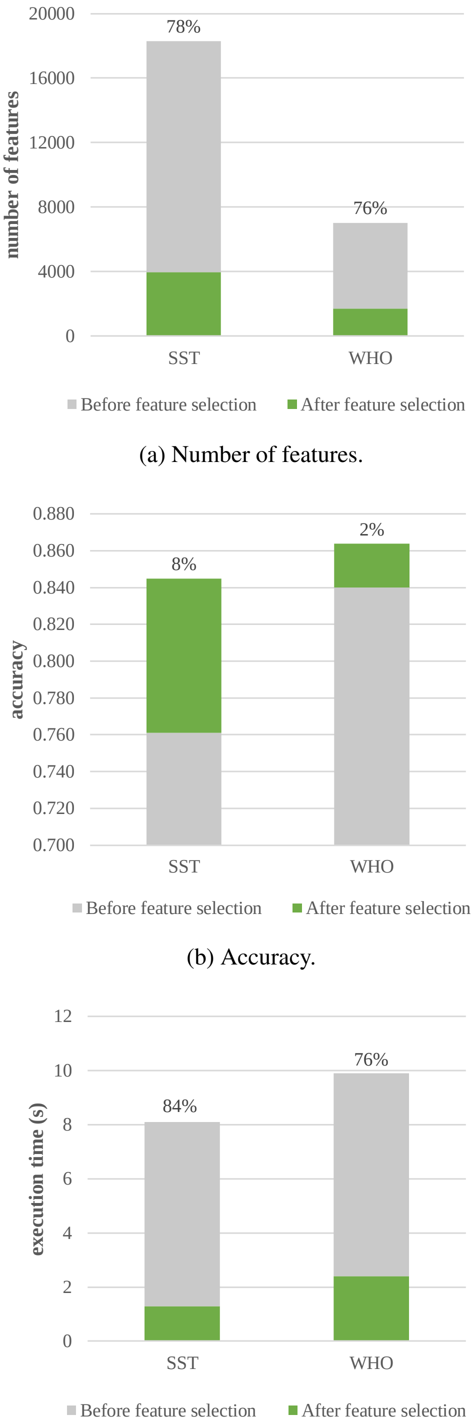
(a) The SST dataset.

(b) The WHO dataset.

FIGURE 5: The initial population and the non-dominated solutions in the final population of the datasets.

we compare our results with off-the-shelf feature selection methods [65]. Table 9 presents the accuracy results for seven well-known feature selection methods along with the proposed model's accuracy with BoW. The feature size parameter of these methods is set the same as our proposed model (e.g., 3972 for LR in SST dataset) to obtain a fair comparison. The results show that the proposed model outperforms all feature selection methods in both datasets regardless of the machine learning technique. Moreover, we implemented a well-known optimization algorithm, i.e., Particle Swarm Optimization (PSO) [66], and compared our results. PSO achieved an accuracy of 0.783 with 8487 features when applied with BoW and LR on the SST dataset. Our proposed model dominated PSO with an accuracy of 0.845 with 3972 features in terms of both accuracy and the number of features. For the WHO dataset, the outcome is similar. PSO was able to achieve an accuracy of 0.861 with 3299 features. The proposed model outperformed it with an accuracy of 0.864 with 1704 features.

Finally, in Table 10, we provide the accuracy results of different methods for the SST dataset presented in the literature. It can be seen from the table that our model (the last row in



(a) Number of features.

(b) Accuracy.

(c) Execution time.

FIGURE 6: The improvements in the number of features, accuracy, and execution time after the proposed algorithm is applied.

TABLE 9: Accuracy comparison with off-the-shelf feature selection methods.

method	SST		WHO	
	LR	SVM	LR	SVM
Fisher score	0.759	0.759	0.851	0.850
ReliefF	0.691	0.692	0.839	0.839
Trace ratio	0.733	0.734	0.850	0.850
Chi-square	0.734	0.734	0.851	0.851
F-statistics	0.733	0.734	0.849	0.850
Gini index	0.764	0.758	0.850	0.850
T-score	0.736	0.740	0.854	0.853
Proposed model	0.845	0.836	0.864	0.861

TABLE 10: Accuracy comparison with conventional methods on the SST dataset.

method	accuracy
SVM [60]	79.4%
NBOW [67]	80.5%
BiNB [68]	83.1%
IWV [69]	83.7%
BOW [70]	80.7%
IGF + NSGA-II	84.5%

the table) achieves better results and proves to be a promising method to enhance the performance of the sentiment analysis task.

C. DISCUSSION

There exist many optimization algorithms for feature selection; however, the skills of these algorithms may change based on the problem they are applied to. According to the No Free Lunch theorem [71], there is no superior algorithm that prevails over every other algorithm in every domain. In this study, we developed a new multiobjective feature selection algorithm for the sentiment analysis domain. We compared our results with many other methods, including conventional methods, off-the-shelf feature selection algorithms, and another optimization algorithm, i.e., Particle Swarm Optimization. We were able to obtain promising results.

Our proposed model decreased the number of features from 18296 to around 4000 for the SST dataset and 7028 to around 1700 for the WHO dataset with the BoW representation. In BoW, the informative words are selected with the feature selection process as the features are the words. Therefore, the sentiment-oriented vocabulary of the dataset is decided with this representation. The classification accuracy increased by around 8% and 2% with this sentiment-oriented vocabulary for the SST and WHO datasets, respectively. Similar to BoW, the proposed model decreased the number of features significantly and increased the accuracy noticeably with the GloVe representation. However, the semantics of feature selection with these two representations are different. A word embedding represents each word with a vector of latent features. Therefore, each dimension of the vector carries different hidden information. In GloVe, each dimension of the 50-dimensional word vectors represents one feature in our study. In addition, since the vectors are concatenated based on the words' order in the sentence, the word's position in the sentence also becomes important. As a result, the algorithm may select a different number of features from different word positions in the sentences to improve the sentiment classification performance. With this approach, our model infers which words and their hidden features contribute more to the sentiment classification task. Moreover, representing texts with word embeddings has become a de facto standard in the NLP literature [23]. Once sentences are built using word embeddings, they are fed into deep learning architectures, such as Convolutional Neural Networks or Long-Short Term Memory networks, as input. These networks determine the weights of each feature in the input separately; hence, possibly approximating weights of some features to zero. Even though our model does not utilize a neural network architecture, it employs a similar idea and nullifies the weights of nonselected features.

There are many reasons why our proposed algorithm can obtain competitive results. Even though evolutionary algorithms evolve through generations and approximate the optimal solution, their computation cost increases excessively as the chromosome size increases. NLP tasks, such as sentiment analysis, are known to have enormous data sizes. As we target to improve the sentiment classification task, we employ an intelligent technique, i.e., filter-based feature selection based on information gain values, on our data before we run our evolutionary algorithm. With this approach, we shrink the chromosome size for our evolutionary algorithm, which boosts the performance in return. In addition, many algorithms depend on an extensive parameter tuning step to achieve better results. On the other hand, our proposed model does not rely on parameter tuning before execution, making it a compelling approach for sentiment classification problems.

In a nutshell, we propose a hybrid feature selection model for the sentiment analysis task. We present many execution results with different feature extraction techniques, optimization algorithms, and machine learning techniques on datasets having different characteristics. These results show that our model is generic, i.e., it works well regardless of the execution setting.

V. CONCLUSION

In this paper, we proposed a hybrid multiobjective feature selection algorithm to improve the performance of the sentiment classification task in various domains. Our model combines a filter-based approach based on the Information Gain metric and a wrapper-based approach based on the Non-dominated Sorting Genetic Algorithm II. We held experiments with the well-known SST benchmark dataset and a real-world dataset we have formed using the speeches of the Director-General of WHO during the COVID-19 pandemic. Experiment results showed that our proposed model significantly improved learning performance. It increased the accuracy by up to 8% and decreased the number of features by up to 78% over baseline sentiment classification models, which eventually reduced computation time and space. We presented the progression of our algorithm using both textual and visual representation of the results in a multiobjective fashion, including both accuracy and feature size. Moreover, we verified the effectiveness of our model by comparing our results with off-the-shelf feature selection techniques and conventional methods applied on the benchmark dataset, including a well-known optimization algorithm, i.e., Particle Swarm Optimization. The results showed that the proposed model is promising to improve sentiment classification performance in datasets of different domains in terms of accuracy and computation costs by selecting the most informative features.

In future work, we plan to enhance our feature selection model by combining different metaheuristic optimization algorithms, such as Particle Swarm Optimization and Krill Herd Optimization. We also aim to build a feature selection model that controls the feature vectorization step, favoring the sentiment analysis's performance. Moreover, we intend to evaluate the performance of the model for different machine learning algorithms and on more datasets from different domains.

REFERENCES

[1] J. A. Morente-Molinera, G. Kou, K. Samuylov, R. Ureña, and E. Herrera-Viedma, "Carrying out consensual group decision making processes under social networks using sentiment analysis over comparative expressions," *Knowledge-Based Systems*, vol. 165, pp. 335–345, 2019.

[2] H. Zhuang, F. Guo, C. Zhang, L. Liu, and J. Han, "Joint aspect-sentiment analysis with minimal user guidance," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1241–1250.

[3] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, S. Seth, and Shubham, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, pp. 26 172–26 189, 2020.

[4] X. Xu, "What are customers commenting on, and how is their satisfaction affected? examining online reviews in the on-demand food service context," *Decision Support Systems*, p. 113467, 2020.

[5] T. B. Mirani and S. Sasi, "Sentiment analysis of isis related tweets using absolute location," in *2016 International Conference on Computational Science and Computational Intelligence*. IEEE, 2016, pp. 1140–1145.

[6] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment analysis of twitter data during critical events through bayesian networks classifiers," *Future Generation Computer Systems*, vol. 106, pp. 92 – 104, 2020.

[7] D. Won, Z. C. Steinert-Threlkeld, and J. Joo, "Protest activity detection and perceived violence estimation from social media images," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 786–794.

[8] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: aggregating and visualizing microblogs for event exploration," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2011, pp. 227–236.

[9] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 system demonstrations*, 2012, pp. 115–120.

[10] T. Renault, "Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages," *Digital Finance*, vol. 2, pp. 1 – 13, 2020.

[11] L. Abualigah, H. E. Alfar, M. Shehab, and A. M. A. Hussein, "Sentiment analysis in healthcare: A brief review," in *Recent Advances in NLP: The Case of Arabic Language*, 2020, pp. 129–141.

[12] A. Alamoodi, B. Zaidan, A. Zaidan, O. Albahri, K. Mohammed, R. Malik, E. Almahdi, M. Chyad, Z. Tareq, A. Albahri, H. Hameed, and M. Alaa, "Sentiment analysis and its applications in fighting covid-19 and infectious diseases: A systematic review," *Expert Systems with Applications*, p. 114155, 2020.

[13] Y. Noh, S. Park, and S.-B. Park, "Aspect-based sentiment analysis using aspect map," *Applied Sciences*, vol. 9, no. 16, p. 3239, 2019.

[14] P. Karagoz, B. Kama, M. Ozturk, I. H. Toroslu, and D. Canturk, "A framework for aspect based sentiment analysis on turkish informal texts," *Journal of Intelligent Information Systems*, vol. 53, no. 3, pp. 431–451, 2019.

[15] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, 2020.

[16] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[17] A. Rasool, R. Tao, M. Kamyab, and S. Hayat, "Gawa–a feature selection method for hybrid sentiment classification," *IEEE Access*, vol. 8, pp. 191 850–191 861, 2020.

[18] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information sciences*, vol. 181, no. 6, pp. 1138–1152, 2011.

[19] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[20] D. Li, R. Rzepka, M. Ptaszynski, and K. Araki, "Hemos: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media," *Information Processing & Management*, vol. 57, no. 6, p. 102290, 2020.

[21] M. Usama, B. Ahmad, E. Song, M. S. Hossain, M. Alrashoud, and G. Muhammad, "Attention-based sentiment analysis using convolutional and recurrent neural network," *Future Generation Computer Systems*, vol. 113, pp. 571–578, 2020.

[22] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, "Understanding emotions in text using deep learning and big data," *Computers in Human Behavior*, vol. 93, pp. 309–317, 2019.

[23] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[24] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.

[25] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[26] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[27] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University-Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[28] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," in *Computer Science '98 Proceedings of the 21st Australasian Computer Science Conference*, February 1998, pp. 181–191.

[29] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[30] H. E. Kiziloz, "Classifier ensemble methods in feature selection," *Neurocomputing*, vol. 419, pp. 97 – 107, 2021.

[31] S. Maldonado, R. Weber, and F. Famili, "Feature selection for high-dimensional class-imbalanced data sets using support vector machines," *Information sciences*, vol. 286, pp. 228–246, 2014.

[32] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," in *Proceedings of the 2012 ACM research in applied computation symposium*, 2012, pp. 1–7.

[33] A. Madasu and S. Elango, "Efficient feature selection techniques for sentiment analysis," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6313–6335, 2020.

[34] A. Bommert, X. Sun, B. Bischl, J. Rahnenführer, and M. Lang, "Benchmark for filter methods for feature selection in high-dimensional classification data," *Computational Statistics & Data Analysis*, vol. 143, p. 106839, 2020.

[35] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, "Ant colony optimization for text feature selection in sentiment analysis," *Intelligent Data Analysis*, vol. 23, no. 1, pp. 133–158, 2019.

[36] M. Tubishat, M. A. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in arabic sentiment analysis," *Applied Intelligence*, vol. 49, no. 5, pp. 1688–1707, 2019.

[37] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A.-Z. Ala'M, I. Aljarah, and H. Faris, "An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on twitter," *Knowledge-Based Systems*, vol. 192, p. 105353, 2020.

[38] S. Shayaa, N. I. Jaafar, S. Bahri, A. Sulaiman, P. S. Wai, Y. W. Chung, A. Z. Piprani, and M. A. Al-Garadi, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37 807–37 827, 2018.

[39] Z. Wang and Z. Lin, "Optimal feature selection for learning-based algorithms for sentiment classification," *Cognitive Computation*, vol. 12, no. 1, pp. 238–248, 2020.

[40] S. R. Ahmad, A. A. Bakar, and M. R. Yaakub, "Metaheuristic algorithms for feature selection in sentiment analysis," in *2015 Science and Information Conference (SAI)*. IEEE, 2015, pp. 222–226.

[41] L. Shang, Z. Zhou, and X. Liu, "Particle swarm optimization-based feature selection in sentiment classification," *Soft Computing*, vol. 20, no. 10, pp. 3821–3834, 2016.

[42] A. Kumar and R. Khorwal, "Firefly algorithm for feature selection in sentiment analysis," in *Computational Intelligence in Data Mining*. Springer, 2017, pp. 693–703.

[43] O. Gokalp, E. Tasci, and A. Ugur, "A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification," *Expert Systems with Applications*, vol. 146, p. 113176, 2020.

[44] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[45] T. Bhattacharyya, B. Chatterjee, P. K. Singh, J. H. Yoon, Z. W. Geem, and R. Sarkar, "Mayfly in harmony: A new hybrid meta-heuristic feature selection algorithm," *IEEE Access*, vol. 8, pp. 195 929–195 945, 2020.

[46] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125 076–125 096, 2020.

[47] H. E. Kiziloz, A. Deniz, T. Dokeroglu, and A. Cosar, "Novel multiobjective tlbo algorithms for the feature subset selection problem," *Neurocomputing*, vol. 306, pp. 94–107, 2018.

[48] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, "Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection," *IEEE Access*, vol. 8, pp. 121 127–121 145, 2020.

[49] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 874–888, 2020.

[50] Y. Zhang, S. Cheng, Y. Shi, D.-w. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.

[51] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, and X.-y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67–85, 2020.

[52] G. Ansari, T. Ahmad, and M. N. Doja, "Hybrid filter–wrapper feature selection method for sentiment classification," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9191–9208, 2019.

[53] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Twitter sentiment analysis using hybrid cuckoo search method," *Information Processing & Management*, vol. 53, no. 4, pp. 764–779, 2017.

[54] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, 2014.

[55] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

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

[57] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[58] T. Kucukyilmaz, A. Deniz, and H. E. Kiziloz, "Boosting gender identification using author preference," *Pattern Recognition Letters*, vol. 140, pp. 245–251, 2020.

[59] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[60] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a

sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[61] W. H. Organization, "Who coronavirus (covid-19) dashboard," https://covid19.who.int/, accessed: 2021-09-03.

[62] M. Y. Kiang, "A comparative assessment of classification methods," *Decision support systems*, vol. 35, no. 4, pp. 441–454, 2003.

[63] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[64] A. Deniz, H. E. Kiziloz, T. Dokeroglu, and A. Cosar, "Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques," *Neurocomputing*, vol. 241, pp. 128–146, 2017.

[65] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 94, 2018.

[66] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2012.

[67] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Jun. 2014, pp. 655–665.

[68] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd annual meeting of the association for computational linguistics*, 2015, pp. 1681–1691.

[69] S. M. Rezaeinia, A. Ghodsi, and R. Rahmani, "Improving the accuracy of pre-trained word embeddings for sentiment analysis," *CoRR*, vol. abs/1711.08609, 2017.

[70] J. Barnes, E. Velldal, and L. Øvrelid, "Improving sentiment analysis with multi-task learning of negation," *Natural Language Engineering*, p. 1–21, 2020.

[71] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

AYÇA DENIZ received her B.S. degree in Computer Engineering from TOBB University of Economics and Technology (TOBB ETU), Ankara, Turkey, in 2012 and her M.S. degree in Computer Engineering from Middle East Technical University (METU), Ankara, Turkey, in 2016. She is currently pursuing a Ph.D. degree in Computer Engineering at METU.

From 2013 to 2015, she worked as a software engineer at TOBB ETU, followed by a three-year research assistantship at TED University, Ankara, Turkey. Between 2018 and 2020, she was a software engineer at The Open University, Milton Keynes, UK. She is currently a doctoral researcher at the MA-Computational Social Science Lab at Koç University, Istanbul, Turkey. Her research interests include machine learning, multiobjective optimization, evolutionary algorithms, and natural language processing.

MERIH ANGIN is an Assistant Professor at the College of Administrative Sciences and Economics of Koç University, where she lectures at the Computational Social Sciences graduate program and International Relations Department. Previously, she was a Postdoctoral Fellow at the Weatherhead Center for International Affairs of Harvard University, a postdoctoral research fellow at the Blavatnik School of Government of the University of Oxford, and a visiting scholar at the Mortara Center for International Studies of the Edmund A. Walsh School of Foreign Service at Georgetown University. She holds a Ph.D. degree from the Graduate Institute of International and Development Studies (IHEID), an M.Sc. degree in International Relations from METU, and a Bachelor's degree in Economics from Bilkent University. Her research interests lie in the areas of international organizations, computational social science including agent-based modeling, machine learning, and artificial intelligence (AI) governance. Her research on AI usage in social sciences has been awarded the EU's Marie Skłodowska-Curie Actions Individual Fellowship, as well as the Scientific and Technological Research Council of Turkey's International Fellowship for Outstanding Researchers. In October 2019, she founded the MA-Computational Social Science Lab (MA-CSSL), which is an interdisciplinary research laboratory bringing together researchers from multiple disciplines including political science, economics and computer science.

PELIN ANGIN is an Assistant Professor of Computer Engineering at Middle East Technical University. She completed her B.S. in Computer Engineering at Bilkent University in 2007 and her Ph.D. in Computer Science at Purdue University, USA in 2013. Between 2014-2016, she worked as a Visiting Assistant Professor and Postdoctoral Researcher at Purdue University. Her research interests lie in the fields of cloud computing, IoT security, distributed systems, 5G networks, data mining and blockchain. She is among the founding members of the Systems Security Research Laboratory and an affiliate of the Wireless Systems, Networks and Cybersecurity Laboratory at METU.