A Deep Swarm-Optimized Model for Leveraging Industrial Data Analytics in Cognitive Manufacturing

Akshi Kumar, *Member, IEEE*, and Arunima Jaiswal

***Abstract*—To compete in the current data-driven economy, it is essential that industrial manufacturers leverage real-time tangible information assets and embrace big data technologies. Data classification is one of the most proverbial analytical techniques within the cognitively capable manufacturing industries for finding the patterns in the structured and unstructured data at the plant, enterprise, and industry levels. This article presents a cognition-driven analytics model, CNN-$_{\mathrm{WSA}}$DT, for the real-time data classification using three soft computing techniques, namely, deep learning [convolution neural network (CNN)], machine learning [decision tree (DT)], and swarm intelligence [wolf search algorithm (WSA)]. The proposed deep swarm-optimized classifier is a feature-boosted DT, which learns features using a deep convolution net and an optimal feature set built using a metaheuristic WSA. The performance of CNN-$_{\mathrm{WSA}}$DT is studied on two benchmark datasets and the experimental results depict that the proposed cognition model outperforms the other considered algorithms in terms of the classification accuracy.**

***Index Terms*—Cognitive manufacturing, data classification, deep learning, industrial data, swarm intelligence.**

I. Introduction

WITH the increasing competition globally, the industries have been tackling numerous challenges pertaining to the organization-wide innovation that can generate and accelerate the global value chain network. Data are now a quintessential business asset, and it is revolutionizing the way companies operate across most sectors and industries [1], [2]. Businesses need to employ efficient analytics algorithms that use the quantitative techniques and evidence-based data to facilitate the value creation for practical decision making, business intelligence, improving product quality, and enhancing knowledge management. Cognitive computing is an emerging technology that operates dynamically into a manufacturing process and business environment to yield tangible benefits driven by the Internet of Things (IoT) and analytics. Essentially, the cognitive manufacturing embraces cognitive computing, the industrial IoT, and radical analytics to drive and optimize the manufacturing processes in the recent Industry 4.0 value chain process [3], [4]. It creates a new interaction paradigm between humans and machines that harnesses natural language and sensory-based capabilities to derive the information that enables the manufacturers to make logical, practical, and cognizant decisions. The business outcomes of the cognitive initiatives enable a mass personalization trend, which involves an interplay of services, manufacturing, customer experience, and engineering [5], [6].

The industries worldwide are gradually awakening to this trend and embracing the cognitive manufacturing to leverage the maximum actionable insights from their data, in a bid to improve fundamental business metrics. The digital footprints from multiple industrial data sources, such as data residing across equipment, systems, and processes, are utilized for real-time monitoring and diagnostics, streaming analytics, machine learning, and operations optimization. A substantial volume and variety of data are generated and shared across the IoT, which with the help of the social media has been extensively used in the retail world to benchmark and optimize the product design, its quality, and customer experience. The intelligence-driven solutions for smart manufacturing analytics are highly desirable for improving production and delivery so that the right customer has the right product at the right time. For example, user demographics, preferences, and behaviors can be analyzed for the customized product design and delivery. Analyzing this goldmine of information about customer preferences is an essential stratagem to a responsive, tractable, and predictive smart manufacturing.

Artificial intelligence (AI) technologies, for instance, machine learning and natural language processing (NLP), facilitate a contextual understanding and allow the personalization of products and services for customers. The intelligent adaptive models are required to deal with the information overload on the chaotic and complex social media portals, and to fully realize the benefits of the social media in Industry 4.0 for a connected, optimized, transparent, and proactive marketplace. Certainly, big-data-driven analytics is the key to smart manufacturing, and the user-generated big data is a substantial source for enhancing manufacturing competitiveness.

Data classification is a promising analytic technique, which is extensively used to solve the IoT and big-data-centered problems for various business or personal objectives. Recently, the social sentiment analysis or emotion AI has also been used to determine the insights pertaining to a topic, brand, or event [7]. It is the use of NLP and computational linguistics to interpret and classify online conversations in terms of positive and negative mentions. Concurrently, nature-inspired computing, has also emerged as a new paradigm of the problem solving that mimics a natural system or process to construct analogies and abstractions for solving complex real-world problems. These study how the biological groups, such as ant colonies, beehives, and flocks of birds, react to stimuli, process information, and make decisions. Motivated by the need to embed the analytical capabilities into the core business for the real-time value creation, this article puts forward a novel cognition-driven social media sentiment mining model. The hybrid model is built on the concord of deep learning [convolution neural network (CNN)] and swarm-optimized [wolf search algorithm (WSA)] decision tree (DT) for real-time sentiment analytics. CNNs are the state-of-the-art at feature extraction and have proven helpful to improve the feature representation and feature learning. Simultaneously, DT is an efficient classifier and can be used to improve the classification capability of CNN. The proposed deep swarm-optimized CNN-$_{\mathrm{WSA}}$DT model embraces the pros of both the techniques, where CNN is the automatic feature learner and DT is the sentiment classifier. The CNN-$_{\mathrm{WSA}}$DT has two primary architectural elements, which are as follows.

1) First, a I-D CNN with five layers, namely the embedding layer, convolution layer, activation layer, down-sampling layer (pooling layer), and output layer with the softmax regression, is used to learn the distributed feature vector representations of the input.
2) Second, for the final classification, a DT classifier is used. This DT takes a combination of the learned vector representations from CNN (the output of the top hidden layer) and a metaheuristically optimized feature vector using WSA to finally output the polarity.

The rationale behind this architecture is that the softmax regression (logistic regression), which is customarily used in CNN to output the probabilities of classes, is a weak classifier that often suffers from the difficulty to interpret the results. Moreover, no weightage is given to the relevant features as simply the word embeddings are used for all features. Basically, the softmax distributes the probability 0–1 over the target classes. In the classification, the predictive probabilities obtained at the end of the pipeline (the softmax output) are often erroneously interpreted as the model confidence. It does not express incertitude and may require the calibration of predicted probabilities. That is, a model can be uncertain in its predictions even with a high softmax output. Pertinent studies have reported the use of a strong classifier, such as support vector machine SVM, to perform the final classification as it often produces comparable results to the softmax regression [8]. We opted to replace the softmax layer with DT as primarily it makes the model easy to interpret. Additionally, DT splits the input space into hyperrectangles according to the target and it does not suffer from the imbalanced support vector ratio or soft margin optimization problems, which are commonly observed in the classifiers such as SVM. But on the flip side, DT has a likelihood of reaching a locally optimal solution as it is a top-down algorithm with a divide and conquer approach. Overfitting of the training data can negatively affect the modeling power of the technique and relegate the predictive accuracy. Population-based metaheuristics, especially the ones inspired by nature, have helped in solving different optimization problems and been used successfully for feature selection in many applications. Our previous study in this direction reported the use of the population-based metaheuristic optimization for the optimal feature selection to improve the sentiment classification accuracy [9], [10]. The study also demonstrated that DT was comparable with SVM in terms of the accuracy gain but outperformed SVM with a considerable reduction in the number of features selected. Therefore, in this article, to generate the optimal feature set, we first use the conventional term frequency-inverse document frequency (TF-IDF) feature extraction and then use a metaheuristic optimization algorithm, the WSA, to select the most relevant set of features. WSA imitates the way wolves search the food, survive, and avoid enemies. WSA possesses the individual local searching ability and autonomous flocking movement in tandem [11]. That is, each wolf is an independent hunter with its own behavior and only joins the peer when the peer is in a superior place within its visual range. The hypothesis behind WSA is that rather than looking for the best solution in one direction by forming a single pack/herd, it considers many leaders swarming to the optimal solution from several directions. Also, to avert trapping in the local optima, the appearance of a hunter (threat/enemy) corresponding to each wolf is randomly added such that the wolf escapes from the hunter's visual range to strive for better solutions within in the search space.

Thus, in the proposed CNN-$_{\mathrm{WSA}}$DT model, the DT is trained using a *boosted feature vector* obtained by combining the CNN-trained features and WSA-optimized feature vector. The model is evaluated on two benchmark datasets, SemEval 2016 (DS-I)[1] and SemEval 2017 (DS-II)[2]. The experiments show that the proposed deep swarm-optimized model has a superior sentiment classification accuracy.

The rest of the article is organized as follows. Section II discusses the related work. Section III presents the proposed model followed by Section IV, which provides the experimental results. Finally, Section V concludes this article.

II. Literature Review

Customization, collaboration, convenience, and wireless connectivity are what is driving the digital transformation today [12]–[15]. The adoption of Industry 4.0 is empowering the industrial users to securely leverage the data and analytics for predictive analysis, reduce machine downtime, centralize storage, and monitor assets remotely. Interoperability, security, data analysis and transfer, and the integration of information technology to the operation technology are some of the key challenges in adopting industrial IoT. Recently, some techniques and frameworks for the big-data analytics have been introduced for the industrial

[1][Online]. Available: http://alt.qcri.org/semeval2016/task4/
[2][Online]. Available: http://alt.qcri.org/semeval2017/task4/

applications [4], [16], [17]. The user-generated big data sourced through the social media is one of the prominent sources of the industrial data to comprehend business trends. This big data empowers businesses to discover and interpret customer behavior patterns in terms of buzz alerts and opinions for designing and improving customizable products [7]. A social media sentiment mining is one of the most proverbial application-based solutions facilitating the data-driven decision making. A number of machine learning and deep learning methods and models have been reported in the relevant studies within this domain [7], [9], [18]. Few studies have also reported the use of the metaheuristic-based algorithms to improve the classification accuracy. In 2013, Basari *et al*. [19] applied SVM and particle swarm optimization (PSO) for the sentiment analysis on the dataset taken from Stanford[3] and showed a superlative performance achieved by this combination In 2015, Gupta *et al*. [20] reported the use of conditional random field and PSO on Twitter SemEval corpus for the multiobjective optimization. Results depicted that the proposed method yielded enhanced classification results using the optimization. In 2017, Kumar and Khorwal [21] implemented the SVM, genetic algorithm, and optimization-based firefly algorithm for the sentiment analysis on the Twitter dataset. Amongst all, the proposed application of the firefly algorithm together with SVM yielded an improved accuracy of 6%. In our earlier work, we focused on the application of swarm optimizers, namely grey wolf and moth flame algorithms, for the sentiment analysis on Twitter corpus (SemEval 2016 and 2017) [9] and cuckoo search to Kaggle corpus [10]. A considerable accuracy gain and feature set reduction was observed.

It is observed that much amount of work within the domain has already been done using machine learning techniques. The studies also demonstrate that the use of deep learning and swarm-based optimization always enhance the classification with less error rate and higher prediction accuracy. Motivated by this, we aim to build a hybrid learning model tapping the benefits of both the deep learning and swarm-optimization algorithms.

III. Proposed CNN-$_{\text{WSA}}$ DT Model

Integrating social media into IoT has emerged as a tool to support the smart manufacturing especially for the product design. Consumers are connected to the industry via social networks, customer interactions, and data analytics. The proposed deep swarm-optimized classification model proffers an analytical method that facilitates the data-driven smart manufacturing. The architecture of the CNN-$_{\text{WSA}}$ DT model is given in Fig. 1.

It consists of the following three architectural components.

1) CNN for feature learning.
2) WSA for the optimized feature generation.
3) Feature-boosted DT for classification.

The underlying notion that drives this model is that the neural architectures are cognitive because they exhibit intelligent behavior by knowing how to categorize, classify, and remember [22], [23]. Concurrently, swarms are cognitive systems because they know how to forage, find sites, build nests, and even add and subtract small numbers [24], [25].

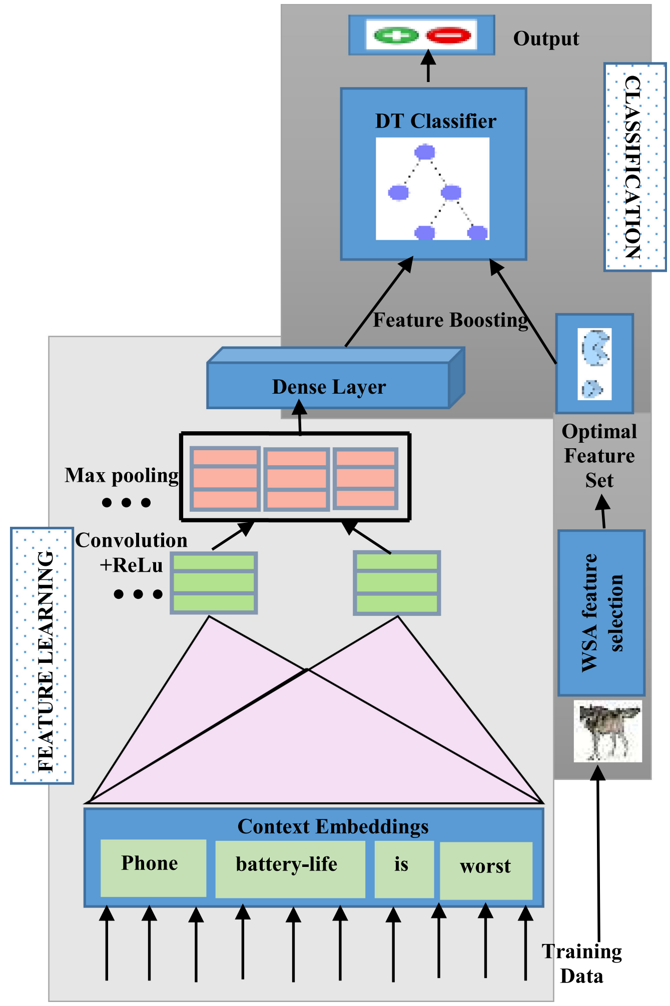

Fig. 1. Architecture of the proposed CNN-$_{\text{WSA}}$ DT model.

The first component involves the defining, initializing, and training of CNN [26]. GloVe [27] is used to generate a "word vector table" with an embedding dimension of 300 and a batch size of 50. The model uses a three-layer convolution architecture with a total of 100 convolution filters each for window size (3, 3). This trains the system to learn the vectors for each word (which would be represented as one hot vector initially) and converts each word to a vector of integers of 300 dimensions. The textual data are now converted into the numerical data for performing convolutions. Padding is used to maintain the fixed input dimensionality feature of CNN, in which zeros are filled in the matrix to get the maximum length amongst all comments in dimensionality. The dropout regularization is set to 0.5 to ensure that the model does not overfit. The default activation function ReLU is applied to the output of the convolution layer introducing nonlinearity into the model, which generates a rectified feature map.

Generic pooling is of varied types, such as max, sum, and average, and is used as a down-sampling strategy in convolution

3.[Online]. Available: http://www.stanford.edu/~alecmgo/cs224n/trainingandtestdata.zip

Algorithm 1: Hybrid Learning Model (CNN+$_{\mathrm{WSA}}$DT).

Input: *Train, Dev, Test, SemEval*- Datasets (2016 and 2017)
Output: *Ac*–Accuracy obtained
1: **Begin**: BuildNet()
2: **Initialize**: InitializeNet(Net)
3: **Repeat while** termination condition is satisfied **do**
4: error←TrainNet(Net, Train, Dev)
5: **End-while**
6: **Select** Feature$_{\mathrm{WSA_opt}}$←WSA(Train, Dev)
7: **Select** Feature$_{\mathrm{CNN_rel}}$←CNN(Train, Dev)
8: Hid$_{\mathrm{Train}}$←GetTopHiddenLayer(Net, Test)
9: Features$_{\mathrm{concat}}$←Feature$_{\mathrm{CNN_rel}}$ + Feature$_{\mathrm{WSA_opt}}$
10: Model$_{\mathrm{DT}}$←DT$_{\mathrm{Train}}$(Features$_{\mathrm{concat}}$)
11: Hid$_{\mathrm{Test}}$←GetTopHiddenLayer(Net, Test)
12: Test$_{\mathrm{concat}}$←Hid$_{\mathrm{Test}}$ + Feature$_{\mathrm{opt}}$
13: Ac←DT$_{\mathrm{Test}}$(Model$_{\mathrm{DT}}$, Test$_{\mathrm{concat}}$)
14: **return** (Ac)

networks. In our model, we use a max pooling, which selects the top-*k* features with respect to the multiple hidden layers in order to retain the most significant sentiment feature information. The output derived from the convolution and pooling layers denotes the high-level features of the input tweets. Thus, the *n*-dimensional representation of the text is finally obtained, which is sent to the output layer for the classification. For the final classification, a DT classifier is used that takes a concatenation of the learned vector representations from CNN (the output of the top hidden layer) and a set of the optimal features generated simultaneously by applying WSA on the training data. That is, a boosted feature vector is used to classify the sentiment, thus typifying a deep swarm-optimized classification model (see Algorithm 1).

Steps 1–5 describe the feature learning using CNN followed by the swarm-optimized feature set generation in step 6. Steps 7–13 explicate the feature-boosted classification. The following subsections present a brief discussion on the techniques used to build the proposed hybrid CNN-$_{\mathrm{WSA}}$DT model.

A. Convolution Neural Network

CNN is a sequence of convolutional layers, interspersed with activation functions. It is a deep neural architecture, which has the power of self-tuning and learning skills by generalizing from the training data. The CNN model enhances the feature extraction in tweets, which improves the generic sentiment analysis task [28]. The proposed CNN model comprises of five layers, namely the embedding layer, convolution layer, activation layer, downsampling layer (pooling layer), and output layer.

The posts from the dataset are preprocessed and input into the embedding layer. The embedding layer of a neural network converts the input from a sparse representation into a distributed or dense representation. In this work, we pretrain the model using the GloVe word embedding. The counts matrix is preprocessed by normalizing the counts and log smoothing them. Thus, this

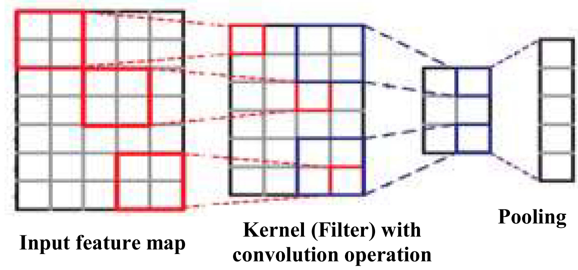

Fig. 2. Convolution operation.

model learns the geometrical encodings (vectors) of the words in each tweet. The proper padding is done for unifying the feature vector matrix. This matrix is given as the input to the convolution layer where a convolution involves a filtering matrix $w \in R^{h*d}$, where h is the size of the convolution, indicating the number of words it spans (see Fig. 2).

The convolution operation is defined as follows:

$$c_i = f\left(\sum\nolimits_{j,k} w_{j,k}\left(X_{[i:i+h-1]}\right)_{j,k} + b\right) \tag{1}$$

where $b \in R$ is a bias term and *f*(*x*) is a nonlinear function, which is the ReLU activation function. Every hidden unit consists of three convolution and max-pooling layers. The output $c \in R^{s'-h+1}$ is, therefore, a concatenation of the convolution operator over all possible windows of the words in the tweet. The activation (ReLU) layer is intended to introduce nonlinearity to the system and produces a rectified feature map, which is inserted into the pooling layer where a max-pooling operation is applied to each convolution $c_{\max} = \max(c)$. The max-pooling operation extracts the "*k*" most important features for each convolution. The output of the final convolution layer, that is, the pooled feature map is a representation of our original input tweet. This representation is then used as an input for the DT classifier, which is combined with the optimized feature set to classify as positive (+1) or negative (−1).

B. Decision Trees

DT forms a tree structure to implement the classification or regression models. It continuously splits the data set into smaller subsets based on a criterion to simultaneously generate the tree incrementally. DTs are fast and easy to code, visualize, manipulate, and explain and allow the results to be interpreted very clearly. Other benefits of using DTs include their application to both numerical and categorical independent variables, efficient handling of the missing values in attributes, and robustness against skewed distributions.

The softmax regression (logistic regression) is generally used in the fully connected output layer of the CNN but has a single linear boundary, unlike DT where we get a nonlinear decision boundary. However, when the classes are not well separated, the trees are susceptible to overfitting the training data. Moreover, tree splitting is locally greedy and the DT is more likely to get stuck in local optima. Therefore, to avert being stuck in the local optimal we use the metaheuristic optimization.

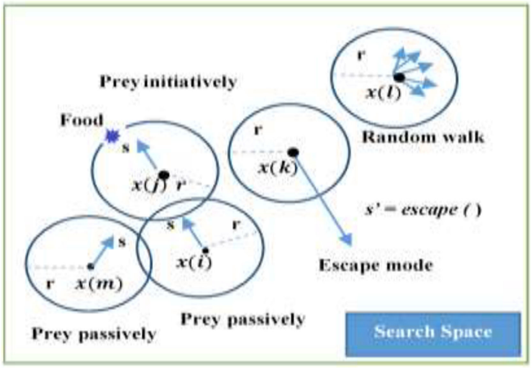

Fig. 3. WSA in action.

C. Metaheuristic Optimization Using WSA

As one of the key subtask in the data classification, feature engineering is the data manipulation process of using the domain knowledge to prepare a compatible dataset for the machine learning algorithm. It includes feature extraction (*n* grams, word2vec, TF-IDF, etc.), feature transformation (scaling, median filling, etc.), and feature selection (statistical approaches, selection by modeling, grid search, and cross validation) [29].

The metaheuristic optimization algorithms have been progressively studied as the wrapper feature selection methods to find the candidate solutions in large search spaces. WSA is formulated by the simulation of the preying behavior of wolves. A wolf in WSA hunts independently and rarely joins its peer, provided the peer has inhabited a better terrain. WSA can be visualized as multiple individual wolves gathering from various directions toward the optimal solution, instead of a single herd searching for the best solution in one direction at a time (see Fig. 3).

The natural behaviors of wolves are simulated in WSA as follows [11].

1) The wolves have an unparalleled memory, which stores the food in caches and track the prey. This unparalleled memory is simulated in WSA, where each wolf has the memory caches that store the positions that are previously visited by it.
2) The wolves search for the prey during hunting and at the same time they watch out for the threats coming toward them. WSA includes a threat probability mechanism that imitates the wolves encounter with enemies. In this condition, the wolf moves away in a random direction by a large distance from its position, which prevents getting stuck in local optima.
3) The wolves have an outstanding judgment of smell, which helps them to locate the prey. WSA simulates this by enabling each wolf to have a sensing distance that creates a coverage area, which is called a visual distance. While searching when a wolf is not able to find the food (the global optimum) or a better terrain than its current position within the visual range, the wolves move in the Brownian motion.

The WSA follows some rules, which are given as follows.

1) The wolves have a visual distance with a radius as v and X as a set of continuous possible solutions. In hyperplane, this distance would be estimated by Minkowski distance as follows:

$$v \le d(x_i,\ x_c) = \left(\Sigma_{k=1}^{n}|x_{i,k} - x_{c,k}|^{\lambda}\right), \quad x_c \in X \tag{2}$$

where x_i is the current position, x_c are all the potential neighboring positions near x_i, and Λ is the order of the hyperspace.

2) The quality of a wolf's current position is given by the fitness of the objective function. The wolf continually attempts to relocate to better terrain inhabited by a companion and will finally choose the best terrain in case of multiple better terrains. Else, the wolf will continue to move randomly in the Brownian motion.
3) When a wolf senses an enemy, it will escape to a random position beyond its visual range to move away from the threat.

1) Merging With Other Wolves: In WSA, the fitness of the objective function determines the quality of the wolf's current position. A wolf always wants to be in a position where there is a greater probability of finding a prey (food) and lower probability of meeting a predator (being hunted), and it will rarely move into the territory occupied by another wolf if that territory is better. This works as follows.

Initially, each wolf locates other wolves within its visual range and evaluates the quality of the position of each of its companions. The best location amongst all is compared with the wolf's position. If it is beneficial to locate to this new position, the wolf relocates and prey there. Otherwise, the wolf searches in a Brownian motion with an incremental step size. The implementation of this movement is given as

$$x(i) = x(i) + \beta_o e^{-r^2}(x(j) - x(i)) + \text{escape}(\,) \tag{3}$$

where escape() generates a random position that enables the wolf to hop, $x(i)$ is the location of the wolf, $x(j)$ is the neighbor that is in a better position, and $\beta_o e^{-r^2}$ is the incentive formula, which represents the betterment (gain) achieved by the wolf by moving to a new position, where β_o is the origin of the food, and r is the distance between the wolf and the new position.

If there are no better terrains occupied by the wolf's peers and the wolf is only in the best position, the other wolves will ultimately crowd to the wolf's current position.

2) Preying: Typically, a wolf looks out a region completely to search for the food in a pattern of the Brownian motion. WSA exhibits three different kinds of preying behavior, which are as follows.

1) Preying initiatively: The objective of the optimization function is represented as a food. In this step, each wolf checks its visual range to detect the prey. The wolf will move step by step in the direction of the prey detected with the highest fitness.
2) Prey passively: In case the wolf is not able to find the food or better position occupied by a peer in the preceding step, it will prey passively by staying alert for the incoming

Algorithm 2: WSA Algorithm [11].

Objective function $f(x), \ \ x = (x_1,\ x_2,\ x_3, \ldots, x_d)^T$
Initialize the population of wolves
$x_i\ (i = 1,\ 2,\ 3, \ldots,\ W)$
Define and initialize parameters:
$r =$ radius of the visual range
$s =$ step size by which a wolf moves at a time
$\alpha =$ velocity factor of the wolf
p_a a user-defined threshold $[0 \ldots 1]$, determines how frequently an enemy appears
While ($t <$ generations && stopping criteria not met)
For $I = 1$: W //for each wolf
Prey_new_food_initiatively ();
Generate_new_location (); //check whether the next location suggested by the random number generator is new. If not, repeat generating random location
If $\mathrm{dist}(x_i,\ x_j) < r$ && $x_j$is better as $f(x_i) < f(x_j)$
x_i moves towards x_j //x_j is better than x_i
Else-if
$x_i =$ *Prey_new_food_passively ();*
End-if
Generate_new_location ();
If $\mathrm{rand}() > p_a$
$x_i = x_i + \mathrm{rand}() + v$; //escape to a new position
End-if
End-for
End-while

threats and also it will check the position of its peers in an attempt to improve its current position.

3) Escape: The wolf escapes quickly when a threat is detected. It relocates itself to a random new position such that its escape distance is greater than its visual range. Escape prevents all the wolves from getting stuck at a local optimum.

These preying steps can be defined mathematically given as follows:

$$\text{if moving} = \begin{cases} x\,(i) = x\,(i) + \alpha \cdot r \cdot \mathrm{rand}() & //\text{Prey} \\ x\,(i) = x\,(i) + \alpha \cdot s \cdot \mathrm{escape}() & //\text{Escape} \end{cases} \tag{4}$$

where $x(i)$ is the position of the wolf, α is the velocity, $\mathrm{rand}()$ is a random function with the mean value in [−1,1], *v* is the visual distance, and *s* is the step size. $\mathrm{escape}()$ is a custom function that generates a position in a random manner, which is greater than *v* and less than half of the solution boundary.

Algorithm 2 describes the WSA.

The parameters for WSA were set as: population size = 20; iterations = 20; chaotic coefficient = 4.

IV. Results and Discussion

The proposed CNN-$_{\mathrm{WSA}}$DT was evaluated for classification performance accuracy Ac (in percentage). Two benchmark Twitter datasets, SemEval 2016 (Task 4, subtask-*A*) and SemEval 2017 (Task 4, subtask-*A*), were used for the training and validation. The tweets were labeled as positive, negative, and neutral. Both SemEval 2016 and 2017 are unbalanced datasets with the SemEval 2017 dataset (DS-II) comprising of 2352 positive, 3811 negative, and 5742 neutral tweets, and the SemEval 2016 dataset (DS-I) consisting of 7059 positive, 3231 negative, and 10 341 neutral tweets. The classification results were assessed by partitioning the dataset into training and test sets. Tenfold cross validation was performed to create a validation set and find the best parameters. We used the Scikit-learn library and Keras deep learning library with the Theano backend.

TABLE I
Results of CNN + $_{\mathrm{WSA}}$DT

Data Set	Accuracy (Ac)
DS-I	89.4%
DS-II	89.7%

TABLE II
Comparative Analysis of CNN and Hybrid Model for DS-I and II

	CNN	CNN+$_{\mathrm{WSA}}$DT
DS-I	85.9	89.4
DS-II	86.1	89.7

A. Performance of the Proposed CNN + $_{WSA}$DT

The accuracy reported by the hybrid model was approximately 90% for both the datasets. This is primarily because CNN does not depend on the extensive manual feature engineering. It employs an automatic extensive feature extraction mechanism. This aids in learning and modeling the real-world problems more efficiently and thus realizing a robust, dynamic, and flexible deeper neural architecture. Also, the application of the WSA optimization produced a set of optimized features, which were combined with the pooled feature of CNN to train the DT classifier for the improved classification accuracy. Table I depicts the accuracy results achieved.

To highlight the improvement shown by the proposed model, we evaluated the CNN model [28] independently as a baseline on both the datasets. The proposed CNN+$_{\mathrm{WSA}}$DT model achieves nearly 3.5% more prediction accuracy, as given in Table II.

Fig. 4 depicts the AUC-ROC curves for DS-I and DS-II.

B. Comparison of DT With Other Supervised Machine Learning Techniques

To endorse the use of DT, it was compared with four other supervised machine learning techniques, namely, SVM, naïve Bayes (NB), *k*-nearest neighbor (*k*-NN), and multilayer perceptron (MLP), on both the datasets. The TF-IDF weighting [30] was used to construct the features set used to train the classifiers. SVM achieved the highest accuracy with 63% for DS-I and 65% for DS-II. MLP also depicted encouraging results with an accuracy of around 60% and 61% for DS-I and DS-II, respectively. Next to MLP, DT attained an accuracy of 55% and 59% for DS-I and DS-II, respectively. Fig. 5 depicts the

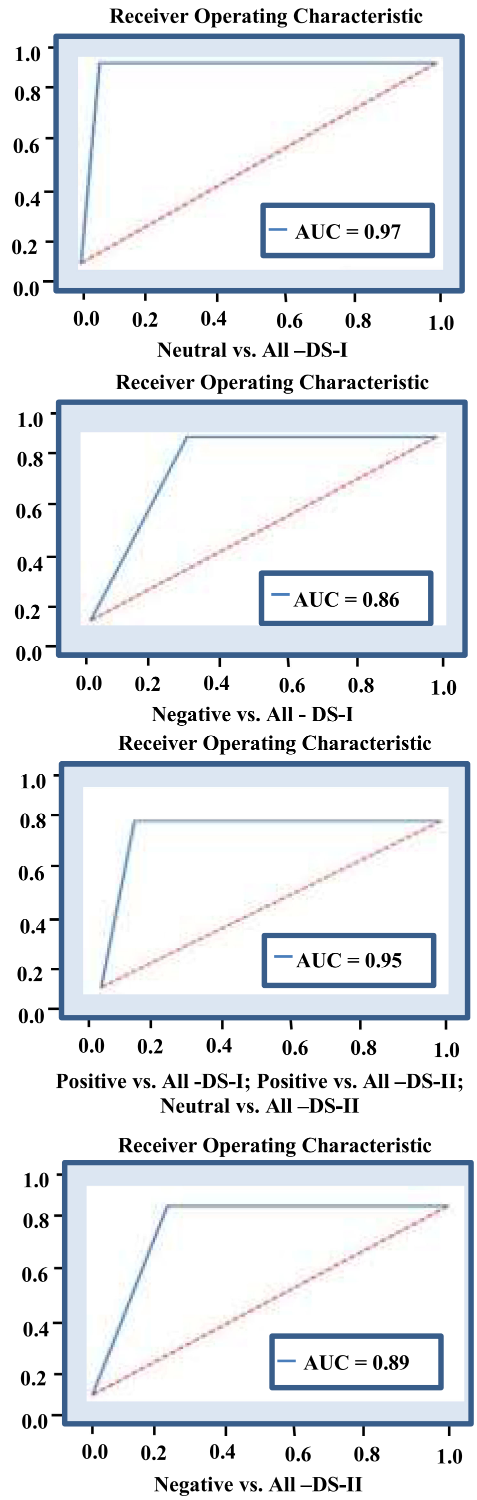

Fig. 4. ROC for DS-I and DS-II.

comparative analysis of the aforesaid techniques based on the accuracy percentage.

Although SVM individually showed the highest accuracy, for the proposed model, we preferred choosing DT so as to develop a robust model for the sentiment analysis, which attune to the “skewness” in real-time datasets. Also, while MLP came next to SVM but as it is a neural model and our model is already using a deep layered neural architecture, the CNN, we opted using DT for the final classification in our proposed hybrid model.

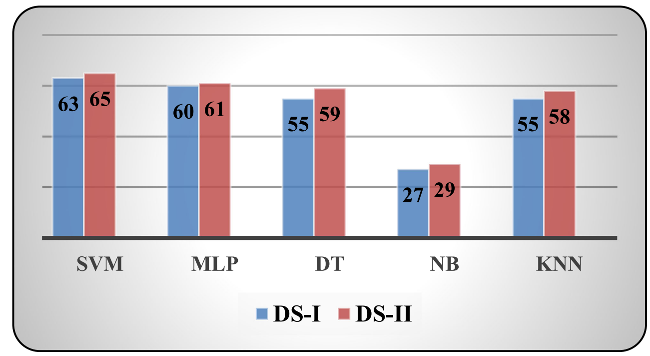

Fig. 5. Comparison of the supervised learning techniques using accuracy (Ac).

TABLE III
FEATURE SELECTION USING TF-IDF+WSA

	DS-I (tf-idf)	(tf-idf + WSA)	%	DS-II (tf-idf)	(tf-idf + WSA)	%
NB	2717	2156	79.35	2658	2156	81.10
DT	2717	2095	77.10	2658	1986	74.71
SVM	2717	2375	87.41	2658	2192	82.46
MLP	2717	2332	85.82	2658	1814	68.24
k-NN	2717	2215	81.52	2658	2070	77.87

Experiments were also done to discern the selection of the optimal subset of features using WSA with these supervised learning techniques. Table III depicts the number and percentage of features selected in both the datasets using five different classifiers and WSA optimization.

Quite clearly, our choice was upheld to DT, which reduced the search space notably by integrating its linearly separable advantage to the nonlinear search capability of WSA. The average feature selection by the WSA-optimized DT was approximately 76% for both the datasets.

C. Comparison of WSA With Other Metaheuristic Optimization Algorithms

We conducted an empirical analysis to validate the use of metaheuristic WSA. That is, five optimization algorithms, namely binary bat algorithm (Bat), binary cuckoo algorithm (Cuckoo), binary grey wolf (BGW), binary moth flame (BMF), and WSA, were used to generate the optimal feature subset and the DT was trained using the optimal feature set to witness the viable classification accuracy. Our recent published works illustrated the use of BGW, BMF [9], and binary cuckoo [10] metaheuristic algorithms for the optimal feature selection based sentiment analysis. Table IV illustrates the comparison of the accuracy achieved for each of the optimization algorithm used to train DT and it is observed that WSA-optimized DT outperforms the others.

TABLE IV
ACCURACY COMPARISON OF OPTIMIZATION ALGORITHMS

	$_{\text{Bat}}$DT	$_{\text{Cuckoo}}$DT	$_{\text{BGW}}$DT	$_{\text{BMF}}$DT	$_{\text{WSA}}$DT
DS -I	63.4	63.3	63.9	64.9	70.2
DS-II	59.1	59.9	69.3	67.7	71.9

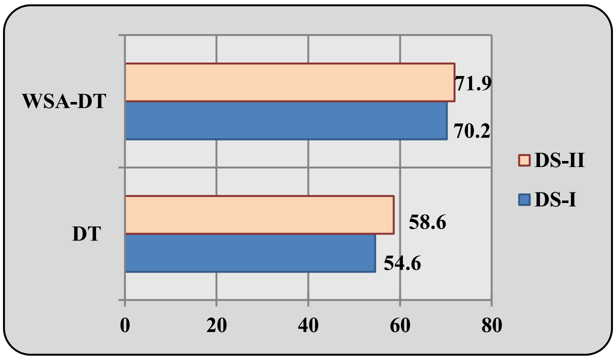

Fig. 6. Accuracy (%) of DT with and without the WSA optimization.

The graph in Fig. 6 shows the accuracy of DT in percentage with and without the WSA optimization.

Thus, an average accuracy gain of 14.5% and an average feature reduction of 24% was observed for DT with the WSA optimization.

V. CONCLUSION

Cognitive manufacturing enables organizations to actively use the advanced analytics to understand, reason, and learn the processes, people, and operations. The cocreation and user experience define this emerging manufacturing paradigm where undeniably a "smart" factory should be able to meet the production needs and deliver customer satisfaction. Based on this, a novel cognition-driven data classification model was put forward in this article, which embedded the predictive analytics capabilities into the core manufacturing task for the real-time value creation. This proposed hybrid model for the real-time sentiment classification used CNN and WSA DT trained and validated on two benchmark Twitter datasets. The combined optimal feature vector generated a superior learning model with an average accuracy of 89.5% validated on both datasets. Ultimately, gauging this user-generated big-data will allow learning context and facilitate a cognitive design for mass personalization. Also, the model added a layer of interpretability and its prospects as an explainable AI solution needs further discussion. As a promising future direction, the CNN + $_{\text{WSA}}$DT can be used to implement the classification algorithm in the programming model, such as MapReduce to achieve parallel processing, thereby solving the problems of hardware and communication overhead for managing large-scale and streaming datasets.

REFERENCES

[1] D. Jiang, Y. Wang, Z. Lv, S. Qi, and S Singh, "Big data analysis-based network behavior insight of cellular networks for industry 4.0 applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1310–1320, Feb. 2020.

[2] V. Jirkovsky, M. Obitko, and V. Marik, "Understanding data heterogeneity in the context of cyber-physical systems integration," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 660–667, Apr. 2017.

[3] P. K. Muhuri, A. K. Shukla, and A. Abraham, "Industry 4.0: A bibliometric analysis and detailed overview," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 218–235, 2019.

[4] A. K. Tripathi, K. Sharma, M. Bala, A. Kumar, V. G. Menon, and A. K. Bashir, "A parallel military dog based algorithm for clustering big data in cognitive industrial internet of things," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2020.2995680.

[5] F. Zhou, Y. Ji, and R. J. Jiao, "Affective and cognitive design for mass personalization: Status and prospect," *J. Intell. Manuf.*, vol. 24, no. 5, pp. 1047–1069, 2013.

[6] P. Zheng, S. Yu, Y. Wang, R.Y. Zhong, and X. Xu, "User-experience based product development for mass personalization: A case study," *Procedia CIRP*, vol. 63, pp. 2–7, 2017.

[7] A. Kumar, K. Srinivasan, W.-H. Cheng, and A.Y. Zomaya, "Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data," *Inf. Process. Manage.*, vol. 57, no. 1, 2020, Art. no. 102141.

[8] Y. Cao, R. Xu, and T. Chen, "Combining convolutional neural network and support vector machine for sentiment classification," in *Proc. Chin. Nat. Conf. Social Media Process.*, Guangzhou, China, Nov. 2015, pp. 144–155.

[9] A. Kumar and A. Jaiswal, "Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on twitter," *Multimedia Tools Appl.*, vol. 78, pp. 29529–29553, 2019.

[10] A. Kumar, A. Jaiswal, S. Garg, S. Verma, and S. Kumar, "Sentiment analysis using cuckoo search for optimized feature selection on Kaggle tweets," *Int. J. Inf. Retrieval Res.*, vol. 9, pp. 1–15, 2019.

[11] R. Tang, S. Fong, X.-S. Yang, and S. Deb, "Wolf search algorithm with ephemeral memory," in *Proc. 7th Int. Conf. Digit. Inf. Manage.*, 2012, pp. 165–172.

[12] Q. Li, M. Wen, B. Clerckx, S. Mumtaz, A. Al-Dulaimi, and R. Q. Hu, "Subcarrier index modulation for future wireless networks: Principles, applications, and challenges," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 118–125, Jun. 2020.

[13] K. Z. Ghafoor *et al.*, "Millimeter-wave communication for internet of vehicles: Status, challenges and perspectives," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2020.2992449.

[14] B. Ji *et al.*, "Survey on the internet of vehicles: Network architectures and applications," *IEEE Commun. Standards Mag.*, vol. 4, no. 1, pp. 34–41, Mar. 2020.

[15] K. S. Keerthi, B. Mahapatra, and V. G. Menon, "Into the world of underwater swarm robotics: Architecture, communication, applications and challenges," *Recent Adv. Comput. Sci. Commun.*, vol. 13, no. 2, pp. 110–119, 2020.

[16] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner, "Scalable distance-based outlier detection over high-volume data streams," in *Proc. IEEE 30th Int. Conf. Data Eng.*, 2014, pp. 76–87.

[17] R. D. Rosa, F. Orabona, and N. Cesa-Bianchi, "The ABACOC algorithm: A novel approach for nonparametric classification of data streams," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 733–738.

[18] A. Kumar and A. Jaiswal, "Systematic literature review of sentiment analysis on Twitter using soft computing techniques," *Concurrency Comput., Pract. Experience*, vol. 32, no. 1, 2020, Art. no. e5107.

[19] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Eng.*, vol. 53, pp. 453–462, 2013.

[20] D. K. Gupta, K. S. Reddy, and A. Ekbal, "PSO-ASent: Feature selection using particle swarm optimization for aspect based sentiment analysis," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.*, 2015, pp. 220–233.

[21] A. Kumar and R. Khorwal, "Firefly algorithm for feature selection in sentiment analysis," in *Proc. Int. Conf. Comput. Intell. Data Mining*, 2017, pp. 693–703.

[22] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, 1982.

[23] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2017.

[24] Z. Reznikova, *Animal Intelligence: From Individual to Social Cognition*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[25] B. Ryabko and Z. Reznikova, "The use of ideas of information theory for studying "language" and intelligence in ants," *Entropy*, vol. 11, no. 4, pp. 836–853, 2009.

[26] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[27] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[28] A. Kumar and A. Jaiswal, "Deep learning based sentiment classification on user-generated big data," *Recent Adv. Comput. Sci., Bentham Sci.*, to be published, doi: 10.2174/2213275912666190409152308.

[29] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Syst. Appl.*, vol. 38, pp. 8144–8150, 2011.

[30] M. P. S. Bhatia and A. Kumar, "Paradigm shifts: From pre-web information systems to recent web-based contextual information retrieval," *Webology*, vol. 7, no. 1, pp. 1–11, 2010.

Akshi Kumar (Member, IEEE) received the Ph.D. degree in computer engineering from the Faculty of Technology, University of Delhi, Delhi, India, in 2011.

She is currently an Assistant Professor with the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India. Her research interests include affective computing, sentiment analysis, and explainable AI.

Arunima Jaiswal is currently working toward the Ph.D. degree in computer science and engineering with the Department of Computer Science and Engineering (CSE), Delhi Technological University, Delhi, India.

She is currently an Assistant Professor with the Department of CSE, Indira Gandhi Delhi Technical University for Women, Delhi, India. Her current research interests are sentiment analysis, social media analytics, and soft computing.