Hindawi
Complexity
Volume 2020, Article ID 1373087, 16 pages
https://doi.org/10.1155/2020/1373087

Research Article

Prediction of Future Terrorist Activities Using Deep Neural Networks

M. Irfan Uddin,[1] Nazir Zada,[2] Furqan Aziz,[2] Yousaf Saeed,[3] Asim Zeb,[4] Syed Atif Ali Shah,[5] Mahmoud Ahmad Al-Khasawneh,[6] and Marwan Mahmoud[7]

[1]*Institute of Computing, Kohat University of Science and Technology, Kohat, Pakistan*
[2]*Center for Excellence in IT, Institute of Management Sciences, Peshawar, Pakistan*
[3]*Department of Information Technology, University of Haripur, Haripur, Pakistan*
[4]*Department of Information Technology, Abbotabad University of Science and Technology, Havelian, Pakistan*
[5]*Faculty of Engineering and Information Technology, Northern University, Nowshehra, Pakistan*
[6]*Faculty of Computer & Information Technology, Al-Madinah International University, Kuala Lumpur, Malaysia*
[7]*King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to M. Irfan Uddin; irfanuddin@kust.edu.pk

Received 31 January 2020; Accepted 1 April 2020; Published 22 April 2020

Academic Editor: Dimitrios Stamovlasis

One of the most important threats to today's civilization is terrorism. Terrorism not only disturbs the law and order situations in a society but also affects the quality of lives of humans and makes them suppressed physically and emotionally and deprives them of enjoying life. The more the civilizations have advanced, the more the people are working towards exploring different mechanisms to protect the mankind from terrorism. Different techniques have been used as counterterrorism to protect the lives of individuals in society and to improve the quality of life in general. Machine learning methods have been recently explored to develop techniques for counterterrorism based on artificial intelligence (AI). Since deep learning has recently gained more popularity in machine learning domain, in this paper, these techniques are explored to understand the behavior of terrorist activities. Five different models based on deep neural network (DNN) are created to understand the behavior of terrorist activities such as is the attack going to be successful or not? Or whether the attack is going to be suicide or not? Or what type of weapon is going to be used in the attack? Or what type of attack is going to be carried out? Or what region is going to be attacked? The models are implemented in single-layer neural network (NN), five-layer DNN, and three traditional machine learning algorithms, i.e., logistic regression, SVM, and Naïve Bayes. The performance of the DNN is compared with NN and the three machine learning algorithms, and it is demonstrated that the performance in DNN is more than 95% in terms of accuracy, precision, recall, and F1-Score, while ANN and traditional machine learning algorithms have achieved a maximum of 83% accuracy. This concludes that DNN is a suitable model to be used for predicting the behavior of terrorist activities. Our experiments also demonstrate that the dataset for terrorist activities is big data; therefore, a DNN is a suitable model to process big data and understand the underlying patterns in the dataset.

1. Introduction

One of the most important threats to today's civilization is terrorism, which has affected the quality of lives of people in the whole world [1]. Terrorism means the use of intentional indiscriminate and illegal power and violence for creating terror amongst general population in order to gain some political, monetary, religious, or legal objectives. The definition of terrorism according to Hoffman [2] is "the deliberate creation and exploitation of fear through violence or the threat of violence in the pursuit of political change." The objectives of terrorism are to create instability by creating fear, anxiety, and uncertainty on a larger scale compared to a single individual. According to Global Terrorism Database (GTD), in 2019 alone 1,411 different terrorist attacks have happened, causing 6,362 fatalities and badly affecting the quality of life of individuals in the society. A visualization of world map showing different terrorist activities is given in Figure 1 (image source: https://www.start.umd.edu/gtd/). The orange color shows high intensity value as a combination of incident fatalities and injuries. The map shows a very high rate of terrorism in South Asia and the Middle East.

The response of terrorist events is constant sense of fear, feeling helpless, experiencing fear and anger, and intolerance or aggression towards certain ethnicity or religious groups. It is equally important that the emotional reactions of the population is understood in regard to terrorist events so that we are able to design assistance to effectively help those who are suffering from these issues or they do not react to carry out another terrorist activity as a revenge. Terrorism has been studied for decades to understand the major factors causing the act of terrorism or understanding how to perform counterterrorism or understanding the social and economic effects of terrorism [3, 4]. However, because of the complex nature of terrorism, it is difficult to find an effective solution that can be used as a counterterrorism to protect the lives of individuals. Identification of terrorist ideologies and prediction of future terrorist attacks have been proven to be of great importance and time-consuming process.

Machine learning algorithms have been used recently to study the different factors of terrorism [5, 6]. NN and particularly DNN are getting popularity mainly because of the fact that a huge amount of labelled data is available recently. The advancements in computer technologies [7–9] have been able to create much powerful computer systems to perform the required computation in DNN. In this paper, NN and DNN models are used to make predictions of different factors that lead to terrorist activities. The model is helpful for law enforcement agencies to make prediction before an incident actually happens and potentially causes the loss of precious lives. The predicted factors are explained below.

(i) Suicide: to predict whether a terrorist activity is going to be suicide or not.

(ii) Success: to predict whether a terrorist activity will succeed or not.

(iii) Weapon type: to make a classification of the general type of weapons used in terrorist activity.

(iv) Region: to classify the region that will be targeted by the terrorist activity.

(v) Attack type: to classify the type of attack carried out as a terrorist activity.

These predictions are important to understand in order to perform counterterrorism. Deep learning can make these predictions efficiently and can help law enforcement agencies to devise mechanisms to deal with terrorists and protect the lives of individuals. With the help of these tools, a terrorist activity can be stopped before it can actually happen and make destructions in terms of lives, infrastructure, or law.

The rest of the paper is organized as follows. Related work is explained in Section 2 to highlight the current state-of-the-art research work in the field. Proposed methodology is explained in Section 3. It also gives a detailed analysis of the dataset, and the architectures of NN and DNN used for the prediction of different factors are explained. Results are demonstrated in Section 4, and the paper is concluded in Section 5 with possible future research directions.

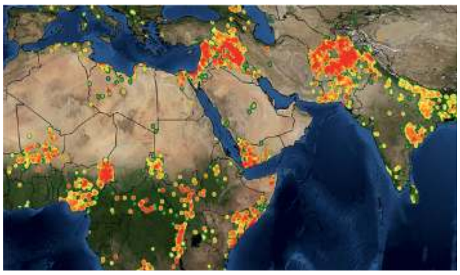


Figure 1: GTD world map highlighting the intensity of terrorism in the year 2017 (focused on South Asia, the Middle East, and Africa).

2. Related Work

Terrorism can affect a society very badly and can have a huge impact on the people. The topic has been studied extensively over the last few decades to understand its causes and how to develop an effective counterterrorism mechanism to reduce the chances of terrorist activities. Machine learning algorithms and data mining techniques have also been applied to understand the different factors involved in a terrorist activity. In 2004, an adaptive safety analysis and monitoring (ASAM) system developed by Singh et al. [10]at University of Connecticut was discussed. The system used hidden Markov models (HMMs) and Bayesian networks (BNs). The system can detect, track, and predict the potential terrorist activities in real time. The paper has demonstrated the use of the ASAM in analyzing the vulnerabilities at the Athens 2004 Olympics. In 2004, Tranchita et al. had developed a classification model in [11] that includes internal and external, natural and unnatural or man caused events. They have developed a new security analysis methods that predicts events uncertainties.

In [12], Godwin et al. developed a visual analytical approach to effectively identify related entities such as terrorist groups, events, and location based on a 2D layout. The paper demonstrates a sequence comparison from bioinformatics, modified to incorporate the element of time. The paper has claimed that the system reveals relationships between entities that are not easily detectable using traditional methods. In 2009, Ozgul et al. [13] proposed an ensemble framework that can classify and predict terrorist groups using four different classifiers: Naïve Bayes, K-NN, Iterative Dichotomiser 3, and decision stump. The authors demonstrated that ensemble framework has better figures compared to individual models. In 2011, Dixon et al. [14] developed a neural network-based framework for counterterrorism. The authors used a game that is designed by criminologist and psychologists to generate data that can test the suitability of AI techniques to look for counterterrorism. The authors investigated neural network and achieved a 60% success rate to identify deceptive behaviour. In 2014, Pilley [15] predicted terrorist groups using CLOPE algorithm.

In 2016, Toure and Gangopadhyay [16] collected incident data from a real-time system to develop a risk model that calculates the terrorism risk level of different locations. A set of rules was also proposed along with the risk model to make prediction of the future terrorist activities. The paper claims to have an accuracy of up to 96%. In another study by Saha et al. [17] in 2017, the authors predicted attack types, weapon types used, and target types, i.e., type of people where attack is made, using ensemble learning algorithm. The paper has claimed to achieve an accuracy in the range of 79% to 86%. In 2017, Mo et al. [18] focused on the prediction of terrorist events from the GTD with data mining techniques. SVM, Naïve Bayes, and logistic regression were used, and they demonstrated an accuracy of up to 78%. In [19], Ding et al. used machine learning methods (NNET, SVM, and Random Forest) to simulate the risk of terrorist attacks. The model was able to predict the places where terrorist events might occur with a success rate of 96%. In 2017, Garg et al. [20] studied the sentiments and survival of tweets before the terrorist attack on September 18, 2016, on security forces by four different terrorists. Different factors of tweets were taken into account such as last retweet, number of retweets, and number of favorites, which were used to study the sentiments of tweets.

Five different machine learning models, i.e., SVM, ANN, Naïve Bayes, Random Forest, and Decision Trees, were used to make predictions on attack type, attack region, and weapon type in 2018 by Verma et al. [21], reporting an accuracy of around 90%. In 2018, Li et al. [22] predicted the behavior of terrorist groups by presenting a comprehensive framework that uses social network analysis, wavelet transform, and pattern recognition approaches to understand the dynamics of the terrorist group and eventually predict the attack behavior. The paper has claimed that the framework has made accurate prediction of the behavior of the terrorist groups. Zhang et al. [23] in 2018 improved the location recommendation algorithm with multisource factors and spatial characteristics using the data of terrorist attack in Southeast Asia from 1970 to 2016. The model was used to build a spatial risk assessment model of terrorist attacks. The paper has claimed to achieve an accuracy of up to 88%.

In another study by Hao et al. [24], the authors used geospatial statistics that can analyze the spatiotemporal evolution of terrorist attacks in Indo-China. Random Forest is used to predict the risk of terrorist attacks using 15 driving factors. In 2019, Agarwal et al. [25] focused on analyzing the dataset of GTD and made prediction on different factors that might have given a blow to terrorism. Different data mining and machine learning algorithms such as SVM, Random Forest, and logistic regression have been used to understand the dataset and predict different factors such as the success of terrorist attack, the group that was involved in terrorist attack and the effect of different external factors involved in terrorist attack. In 2019, Kalaiarasi et al. [26] developed multiple classifiers to group and predict different terrorist activities using k-NN algorithm and Random Forest techniques. They used the GTD dataset for detection of terrorism. In 2019, Maniraj et al. [27] developed a system that examines the growth or decay of the terrorist groups by the time, location, type of attack, target motives, weapon type, and availability. They analyzed the GTD dataset and used machine learning algorithm that can predict the probability of attacks in different regions. In 2019, Christie in his thesis [28] carried out a study to understand the dynamics of unclaimed terrorism events in Pakistan using machine learning algorithms. They made predictions on terrorist attributes such as attack, target, weapon type, spatial attack, and lethality of attacks. The study made an attempt to match the unattributed terrorist attack to known terrorist groups. In 2019, Ahmad et al. [29] developed a method for detection and classification of social media-based extremist affiliations based on the sentiment analysis. The focus was to classify tweets into two categories: extremist and nonextremist classes. The system uses deep learning-based sentiment analysis to make a classification about the tweets. Other similar studies in 2020 can be found in [30–32].

All previous studies have applied machine learning and deep learning techniques to make AI-based model for terrorism. Current state-of-the-art research papers are based on understanding the pattern of terrorism and have proposed different solutions to analyze factors of terrorism. However, no research work is carried out in order to make prediction of future terrorist activities and predict different factors such as success, suicide, weapon type, attack type, and region. Clearly, there is a research gap for modeling and predicting future terrorist activities using deep learning. This research paper compares the performance of traditional machine learning and deep neural networks and concludes that deep neural network is a suitable model for prediction of future terrorist activities.

3. Proposed Methodology

3.1. Data Analysis. In this section, a detailed analysis of the dataset is given. The preprocessing performed on the dataset is also explained.

3.1.1. Feature Selection. The National Consortium for the Study of Terrorism and Responses to Terrorism (START) has prepared a dataset known as Global Terrorism Database (GTD) (https://www.start.umd.edu/gtd). GTD contains information about terrorist activities from 1970 until 2018, including more than 181,000 different instances of terrorism. In this paper, 34 attributes (some attributes are redundant and hence discarded) are taken for the analysis. These attributes along with description are given in Table 1.

3.1.2. Prediction of Different Factors of Terrorist Activities. The following are different factors that neural network and deep neural network will be trained to learn.

(1) Suicide. This field indicates whether the attack is suicide or not suicide. 1 = "Yes" means that the incident was a suicide attack. 0 = "No" means there is no indication that the

Table 1: The attributes in the dataset along with explanation.

S.No.	Feature	Description
1	iyear	This field contains the year in which the incident occurred
2	imonth	This field contains the number of the month in which the incident occurred
3	iday	This field contains the numeric day of the month on which the incident occurred
4	Extended	1 = "Yes," the duration of an incident extended more than 24 hours; 0 = "No," the duration of an incident extended less than 24 hours
5	Provstate	Name (at the time of event) of the 1st order subnational administrative region
6	Latitude	The latitude of the city in which the event occurred
7	Longitude	The longitude of the city in which the event occurred
8	Specificity	
9	Vicinity	The region in nearby location
10	Crit1	
11	Crit2	
12	Crit3	
13	Doubtterr	
14	Multiple	
15	Natlty1	The nationality of the target that was attacked
16	Propextent	
17	Ishostkid	The hostage of kids
18	Ransom	
19	Country	This field identifies the country or location where the incident occurred
20	City	Name of the city, village, or town in which the incident occurred
21	Gname	The name of the group that carried out the attack
22	Individual	
23	Nkillus	The number of U.S. citizens who died as a result of the incident
24	Nkillter	
25	Nwound	Number of confirmed nonfatal injuries to both perpetrators and victims
26	Nwoundus	The number of confirmed nonfatal injuries to U.S. citizens, both perpetrators and victims
27	Nwoundte	
28	Property	The damage to property
29	Targtype1	The general type of target/victim
30	Suicide	1 = "Yes," the incident was a suicide attack; 0 = "No," there is no indication that the incident was a suicide attack
31	Success	Success of a terrorist strike
32	Weaptype1	General type of weapon used in the incident
33	Region	This field identifies the region code based on 12 regions
34	Attacktype1	The general method of attack and broad class of tactics used

incident was a suicide attack. Dimension of the dataset is $(350, 116 \times 34)$. 90% data is used for training (315,104 instances) and 10% is used for testing (35,012 instances). Both "Yes" and "No" classes have 175,058 instances.

(2) Success. This field indicates the success of a terrorist strike. 1 = "Yes" means that the incident was successful. 0 = "No" means that the incident was not successful. Dimension of the dataset is $(323, 264 \times 34)$. 90% of the dataset is taken as training (290,937 instances) and 10% is taken as testing (32,327 instances). Each class has 161,632 instances.

(3) Weapon Type. This field indicates the general type of weapon used in the incident. In the dataset, 13 different labels are used to represent different type of weapon. These labels are explained below.

(1) Biological
(2) Chemical
(3) Radiological
(4) Left as blank
(5) Firearms
(6) Explosives
(7) Fake weapons
(8) Incendiary
(9) Melee
(10) Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)
(11) Sabotage Equipment
(12) Other
(13) Unknown

Dimension of the dataset is $(1, 109, 112 \times 34)$. 90% of the dataset is taken as training (998,200 instances) and 10% is taken as testing (110,912 instances). Each class has 92,426 instances.

(4) Region. This field indicates 12 different regions. These regions are explained below.

(1) North America
(2) Central America and Caribbean
(3) South America
(4) East Asia

(5) Southeast Asia
(6) South Asia
(7) Central Asia
(8) Western Europe
(9) Eastern Europe
(10) The Middle East and North Africa
(11) Sub-Saharan Africa
(12) Australasia and Oceania

Dimension of the dataset is $(605,688 \times 34)$. 90% of the dataset is taken as training (545,119 instances) and 10% as testing (60,569 instances). Each class has 50,474 instances.

(5) Attack Type. This field indicates the general method of attack and broad class of tactics used. In the dataset, 9 different labels are given and are explained below.

(1) Assassination
(2) Armed assault
(3) Bombing/explosion
(4) Hijacking
(5) Hostage taking (barricade incident)
(6) Hostage taking (kidnapping)
(7) Facility/infrastructure attack
(8) Unarmed assaults
(9) Unknown

Dimension of the dataset is $(95,7242 \times 34)$. 90% of the dataset is used for training (861,517 instances) and 10% is used for testing (95,725 instances). Each class has 88,255 instances.

3.1.3. Text to Numbers. In the GTD dataset, some features are in text format, for instance, group name, country name, etc. It is not possible to process features with text data in NN or DNN. There exist multiple techniques to convert text data to numbers, e.g., TFIDF, Word2Vec, GloVe, One hot encoding, etc. In this paper, LabelEncoder class of sklearn library is used to convert nonnumeric data to numeric data, as the labels are hashable and comparable to numerical labels.

3.1.4. Missing Data. The dataset contains many missing values, i.e., the cell does not contain any data, which results into NaN when processed by NN. Different interpolation techniques can be used to fill the missing data. In this paper, SimpleImputer of sklearn library is used to fill the missing data. We have replaced the missing values by *mean* along each column.

3.1.5. Dealing with Unbalanced Classes. During the analysis of the dataset, it is observed that the data are not balanced in different classes. In some classes, there are more instances, while others have very few instances. NN and DNN trained on unbalanced data are biased [33] towards the classes having more instances. In order to keep the data in balanced form, *SMOTE: Synthetic Minority Oversampling Technique* presented by Chawla et al. in 2002 [34] and later made available as a tool to be used in Python in [35] is used. NN and DNN presented in this paper are trained on balanced data.

3.1.6. Normalization. In GTD, data are in different range. Some columns have values as 0 and 1, while others have values in hundreds or thousands. In this situation, it is difficult for learning algorithm to learn the pattern and converge to a global minimum. Therefore, it is important that before the data are processed by a learning model, the data are normalized, i.e., in the range of 0 to 1 or −1 to 1. In this paper, MinMaxScalar of sklearn library is used, which for each value of the feature subtracts the average of all values and divides it by standard deviation, to convert the data in the range of −1 to 1. The formula of standardization is expressed in equation (1), where X_i are all the samples for a given feature, $\overline{X}$ is the average of all samples by the feature, and s is the standard deviation.

$$Z_i = \frac{X_i - \overline{X}}{s}. \tag{1}$$

3.2. Learning Model. In this section, the learning model used for the prediction of terrorist activities is explained. Two different models are developed. One is based on NN and the other is based on DNN. NN [36–38] is a graph of different nodes to perform computation. These nodes are connected with each other by weighted edges. Some of the nodes are classified as input that takes input features and some of the nodes are known as output nodes that make predictions. During the forward propagation, a matrix of weights is multiplied with input features and eventually makes prediction. We have developed five different models. We will explain the process of learning in one model for suicide prediction. There are 34 features, where 33 are input features and 1 is output feature which classifies whether an attack is suicide or not. In order to perform training, we store all data in a matrix. We have $315,104$ instances of terrorist activities; therefore, the size of the input matrix represented by X is $315,104 \times 33$. In order to train on NN, we need to provide a matrix of the weights with the same size as input features. In case there are 10 units in the first layer, then the size of the weight matrix is 33×10. We initialize these weights randomly using Glorot Uniform initializer. We also need to provide a bias represented by b. The formula of this multiplication is shown in equation (2), where W_1 shows the weights for the hidden layer, b_1 shows the bias, and X represents the input matrix. There is a nonlinear function, ReLU [39], which is computed as $\text{ReLU}(z) = \max(0, z)$.

$$\begin{aligned} Z_1 &= W_1^T \times X + b_1, \\ A_1 &= \text{ReLU}(Z_1). \end{aligned} \tag{2}$$

For the output layer, we multiply the output of the hidden layer with different weights. Suppose we have 10 units in the hidden layer and one unit in the output layer, then the dimension of the weight matrix is 10×1. We also need to add a bias at this layer. The calculation performed at the output layer is shown in equation (3), where W_2 and b_2 show the weight and bias for the output layer and A_1 is the input vector. At the output layer, *sigmoi d* [40] is computed as $\text{sigmoid}(Z) = 1/(1+e^z)$.

$$
\begin{aligned}
Z_2 &= W_2^T \times A_1 + b_2,\\
A_2 &= \text{sigmoid}(Z_2),
\end{aligned}
\tag{3}
$$

$$
\mathscr{L}(A_2, Y) = -\frac{1}{m}\sum_{i=0}^{m} Y_i \log(A_2). \tag{4}
$$

During the training phase of the NN, the prediction is made, represented by A_2 as shown in equation (3). Then, the loss is computed comparing the predicted values with actual values. We are using binary cross-entropy loss as shown in equation (4), where m represents the number of samples and Y shows the actual output values. During the backpropagation process, the derivative of the loss is taken for output layer and hidden layer, and weights are updated using optimization techniques, such as gradient descent [41], gradient descent with momentum [42], RMSProp [43], and Adam [44]. The backpropagation with gradient descent is shown in equation (5), where α is the learning rate, L is the loss function given in equation (4), and W and b are weights and bias. The algorithm of training of NN model for *suici de* prediction, with gradient descent optimization algorithm, is given in Algorithm 1.

A sample architecture of NN is given in Figure 2. The figure shows the input layer, which contains the input data. These data are passed to the hidden layer with W and b as shown in equation (2). The output of the hidden layer is passed to output layer with W and b and computed as given in equation (3). From the output layer, loss function is computed as given in equation (4). During the backpropagation process, weights are updated, taking into account the error of the actual Y and predicted Y, as shown in the following equation:

$$
\begin{aligned}
W_1 &= W_1 - \alpha\frac{\partial\mathscr{L}}{\partial W_1},\\
b_1 &= b_1 - \alpha\frac{\partial\mathscr{L}}{\partial b_1},\\
W_2 &= W_2 - \alpha\frac{\partial\mathscr{L}}{\partial W_2},\\
b_2 &= b_2 - \alpha\frac{\partial\mathscr{L}}{\partial b_2}.
\end{aligned}
\tag{5}
$$

A DNN [45, 46] has more layers than a single-layer NN. Generally, more than two hidden layers are considered as DNN. A much larger DNN has 100 s of layers. For instance, ResNet [47] has 152 layers. DNN has recently shown advancements in different fields and has achieved state-of-the-art accuracy in different applications given in [48–50]. A sample architecture of the DNN is given in Figure 3. The computation of forward propagation in DNN is same as NN. The same computation for one hidden layer is now computed for $L-1$ hidden layers. The output layer L computes the results as given in equation (3). Loss is computed as given in equation (4), giving an error of the actual and predicted Y. During the backpropagation, the values of W and b for each layer are updated using gradient descent optimization algorithm as shown in equation (5).

The algorithm of DNN is given in Algorithm 2. First, initialization of the weights and biases of all layers is made using Glorot Uniform initializer [51]. Then, in the forward propagation the linear function and nonlinear activation (i.e., ReLU [39]) are computed at each layer. In the last layer, binary cross-entropy loss function is used to compute the loss. In case of binary classification, sigmoid [40] activation function is used, and in case of multiclass classification, softmax [52] is used. Then, during backpropagation, the derivate of the loss function with respect to weights and biases is taken at every layer. The weights and biases are updated using gradient descent.

Gradient descent optimization is explained in Algorithms 1 and 2. But gradient descent is a very basic optimization algorithm and is used only to explain the concept. There are more advanced optimization algorithms, such as gradient descent with momentum [42], RMSprop [43], and Adam [44]. In this paper, we are using Adam optimization for the learning process as it is one of the most effective optimization algorithms for training in deep neural networks. Adam optimization can be expressed mathematically in equation (6), where $v_{\mathrm{d}W^{[l]}}^{\text{corrected}}$ stores the exponentially weighted average of past gradients with bias correction for layer l, $s_{\mathrm{d}W^{[l]}}^{\text{corrected}}$ calculates exponentially weighted average of the squares of the past gradients for layer l, $(\beta)_1$ and $(\beta)_2$ are hyperparameters that control the two exponentially weighted averages, α is the learning rate, t counts the number of steps taken for Adam optimization, l means the number of layers, and ϵ is a tiny value that is added to avoid divide by zero error.

$$
\begin{aligned}
v_{\mathrm{d}W^{[l]}} &= \beta_1 v_{\mathrm{d}W^{[l]}} + (1-\beta_1)\frac{\partial\mathscr{L}}{\partial W^{[l]}},\\
v_{\mathrm{d}W^{[l]}}^{\text{corrected}} &= \frac{v_{dW^{[l]}}}{1-(\beta_1)^t},\\
s_{\mathrm{d}W^{[l]}} &= \beta_2 s_{dW^{[l]}} + (1-\beta_2)\left(\frac{\partial\mathscr{L}}{\partial W^{[l]}}\right)^2,\\
s_{\mathrm{d}W^{[l]}}^{\text{corrected}} &= \frac{s_{\mathrm{d}W^{[l]}}}{1-(\beta_1)^t},\\
W^{[l]} &= W^{[l]} - \alpha\frac{v_{\mathrm{d}W^{[l]}}^{\text{corrected}}}{\sqrt{s_{\mathrm{d}W^{[l]}}^{\text{corrected}}} + \varepsilon}.
\end{aligned}
\tag{6}
$$

The main objective of this research work is to explore novel techniques in deep learning to understand different parameters such as suicide, success, weapon type, the type of attack, and regions of attack that lead to a terrorist activity. These factors help law enforcement agencies to create strategies for counterterrorism. The deep learning algorithm

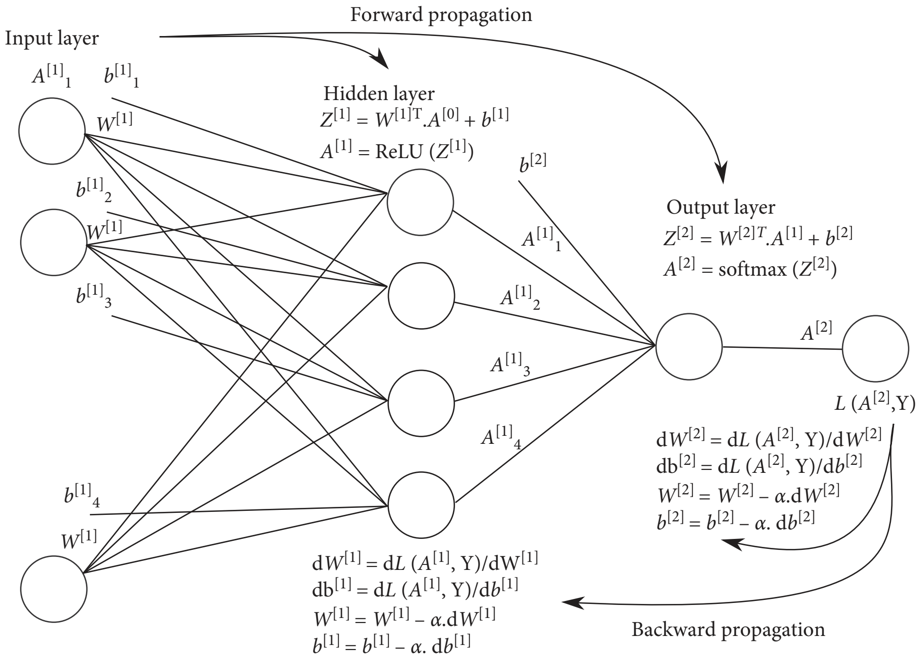

Forward propagation
Input layer
$A^{[1]}{}_1$ $b^{[1]}{}_1$
$W^{[1]}$
$b^{[1]}{}_2$
$W^{[1]}$
$b^{[1]}{}_3$
$b^{[1]}{}_4$
$W^{[1]}$
Hidden layer
$Z^{[1]} = W^{[1]\mathrm{T}}.A^{[0]} + b^{[1]}$
$A^{[1]} = \mathrm{ReLU}\ (Z^{[1]})$
$b^{[2]}$
$A^{[1]}{}_1$
$A^{[1]}{}_2$
$A^{[1]}{}_3$
$A^{[1]}{}_4$
Output layer
$Z^{[2]} = W^{[2]T}.A^{[1]} + b^{[2]}$
$A^{[2]} = \mathrm{softmax}\ (Z^{[2]})$
$A^{[2]}$
$L\ (A^{[2]},\mathrm{Y})$
$\mathrm{d}W^{[2]} = \mathrm{d}L\ (A^{[2]}, \mathrm{Y})/\mathrm{d}W^{[2]}$
$\mathrm{d}b^{[2]} = \mathrm{d}L\ (A^{[2]}, \mathrm{Y})/\mathrm{d}b^{[2]}$
$W^{[2]} = W^{[2]} - \alpha.\mathrm{d}W^{[2]}$
$b^{[2]} = b^{[2]} - \alpha.\ \mathrm{d}b^{[2]}$
$\mathrm{d}W^{[1]} = \mathrm{d}L\ (A^{[1]}, \mathrm{Y})/\mathrm{d}W^{[1]}$
$\mathrm{d}b^{[1]} = \mathrm{d}L\ (A^{[1]}, \mathrm{Y})/\mathrm{d}b^{[1]}$
$W^{[1]} = W^{[1]} - \alpha.\mathrm{d}W^{[1]}$
$b^{[1]} = b^{[1]} - \alpha.\ \mathrm{d}b^{[1]}$
Backward propagation

Figure 2: A sample architecture of neural network.

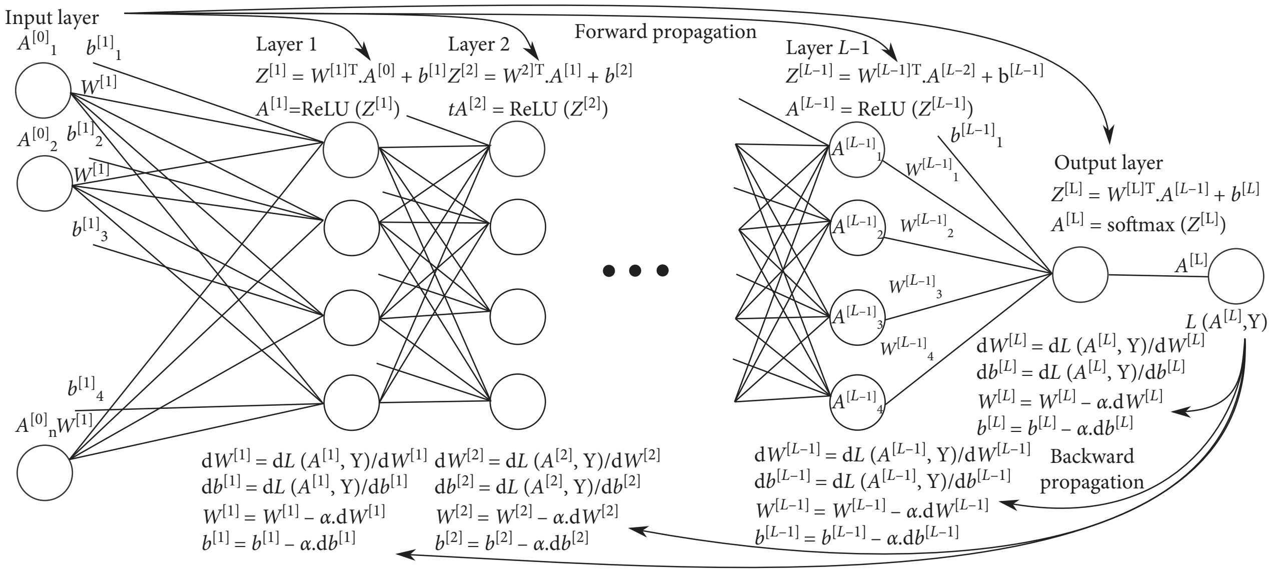

Input layer
$A^{[0]}{}_1$ $b^{[1]}{}_1$
$W^{[1]}$
$A^{[0]}{}_2$ $b^{[1]}{}_2$
$W^{[1]}$
$b^{[1]}{}_3$
$b^{[1]}{}_4$
$A^{[0]}{}_n W^{[1]}$
Layer 1
$Z^{[1]} = W^{[1]\mathrm{T}}.A^{[0]} + b^{[1]}$
$A^{[1]}$=ReLU $(Z^{[1]})$
Layer 2
$Z^{[2]} = W^{2]\mathrm{T}}.A^{[1]} + b^{[2]}$
$tA^{[2]} = \mathrm{ReLU}\ (Z^{[2]})$
Forward propagation
Layer L–1
$Z^{[L-1]} = W^{[L-1]\mathrm{T}}.A^{[L-2]} + \mathrm{b}^{[L-1]}$
$A^{[L-1]} = \mathrm{ReLU}\ (Z^{[L-1]})$
$A^{[L-1]}{}_1$
$A^{[L-1]}{}_2$
$A^{[L-1]}{}_3$
$A^{[L-1]}{}_4$
$b^{[L-1]}{}_1$
$W^{[L-1]}{}_1$
$W^{[L-1]}{}_2$
$W^{[L-1]}{}_3$
$W^{[L-1]}{}_4$
Output layer
$Z^{[\mathrm{L}]} = W^{[\mathrm{L}]\mathrm{T}}.A^{[L-1]} + b^{[L]}$
$A^{[\mathrm{L}]} = \mathrm{softmax}\ (Z^{[\mathrm{L}]})$
$A^{[\mathrm{L}]}$
$L\ (A^{[L]},\mathrm{Y})$
$\mathrm{d}W^{[L]} = \mathrm{d}L\ (A^{[L]}, \mathrm{Y})/\mathrm{d}W^{[L]}$
$\mathrm{d}b^{[L]} = \mathrm{d}L\ (A^{[L]}, \mathrm{Y})/\mathrm{d}b^{[L]}$
$W^{[L]} = W^{[L]} - \alpha.\mathrm{d}W^{[L]}$
$b^{[L]} = b^{[L]} - \alpha.\mathrm{d}b^{[L]}$
Backward propagation
$\mathrm{d}W^{[1]} = \mathrm{d}L\ (A^{[1]}, \mathrm{Y})/\mathrm{d}W^{[1]}$
$\mathrm{d}b^{[1]} = \mathrm{d}L\ (A^{[1]}, \mathrm{Y})/\mathrm{d}b^{[1]}$
$W^{[1]} = W^{[1]} - \alpha.\mathrm{d}W^{[1]}$
$b^{[1]} = b^{[1]} - \alpha.\mathrm{d}b^{[1]}$
$\mathrm{d}W^{[2]} = \mathrm{d}L\ (A^{[2]}, \mathrm{Y})/\mathrm{d}W^{[2]}$
$\mathrm{d}b^{[2]} = \mathrm{d}L\ (A^{[2]}, \mathrm{Y})/\mathrm{d}b^{[2]}$
$W^{[2]} = W^{[2]} - \alpha.\mathrm{d}W^{[2]}$
$b^{[2]} = b^{[2]} - \alpha.\mathrm{d}b^{[2]}$
$\mathrm{d}W^{[L-1]} = \mathrm{d}L\ (A^{[L-1]}, \mathrm{Y})/\mathrm{d}W^{[L-1]}$
$\mathrm{d}b^{[L-1]} = \mathrm{d}L\ (A^{[L-1]}, \mathrm{Y})/\mathrm{d}b^{[L-1]}$
$W^{[L-1]} = W^{[L-1]} - \alpha.\mathrm{d}W^{[L-1]}$
$b^{[L-1]} = b^{[L-1]} - \alpha.\mathrm{d}b^{[L-1]}$

Figure 3: A sample architecture of deep neural network.

was used to learn the pattern of this big data available by GTD using most recent optimization techniques and make reasonable predictions and classifications. Even though many researchers have worked in the domain of using AI solutions for counterterrorism, no one has studied an effective mechanism of understanding factors of terrorism using deep learning, which is becoming very popular recently with the increased data and increased computational [53, 54] power. To the best of the authors' knowledge, no comprehensive work is dedicated to predict and classify factors of terrorism using deep learning algorithms. Therefore, it is sensible to study the problem of predicting future terrorist activities from the perspective of deep learning to demonstrate the full potential of deep neural network.

4. Results

4.1. Experimental Setup on Cluster. The working environment for all experiments in this paper is given in Table 2.

4.2. Architecture of NN and DNN. In the experiments for this paper, the NN consists of one hidden layer having 10 units. The DNN consists of 5 hidden layers. The first layer has 100

```
Input: the whole dataset of GTD along with labels
Output: optimized values of $W$ and $b$
Data: GTD Dataset
$W_1, b_2, W_1, b_2 = random\_numbers$ //Glorot Uniform initializer
while $i \le num\_iteration$ do
    $Z_1 = W_1^T \times X + b_1$
    $A_1 = \mathrm{ReLU}(Z_1)$ //$\mathrm{ReLU}(Z) = \max(0, z)$
    $Z_2 = W_2^T \times A_1 + b_2$
    $A_2 = \mathrm{sigmoid}(Z_2)$ //$\mathrm{sigmoid}(z) = 1/1 + e^z$
    $\mathscr{L}(A_2, Y) = -(1/m) \sum_{i=0}^{m} Y_i \log(A_2)$
    $W_1 = W_1 - \alpha \partial \mathscr{L} / \partial W_1$
    $b_1 = b_1 - \alpha \partial \mathscr{L} / \partial b_1$
    $W_2 = W_2 - \alpha \partial \mathscr{L} / \partial W_2$
    $b_2 = b_2 - \alpha \partial \mathscr{L} / \partial b_2$
```

Algorithm 1: The training of neural network with gradient descent optimization algorithm.

units, second layer has 50 units, third layer has 30 units, fourth layer has 10 units, and fifth layer has 5 units. Both models are trained for 500 epochs, using Adam [44] optimizers implemented in Keras with a learning rate of 0.001. Glorot Uniform initializer [51] is used for the initialization of weights and bias. The different combination of number of layers, number of units per layer, optimization technique, etc., is selected based on the random search in the grid. The model is executed using TensorFlow [55] library and hence exploits the parallelism of the cluster.

4.3. Accuracy in Train and Test Datasets. The accuracy of train and test datasets for every iteration computed by NN and DNN is given in Figure 4. The accuracy at 500 iterations for suicide prediction on NN is shown in Figure 4(a) and that on DNN is shown in Figure 4(b). It can be observed that the accuracy in train on DNN is more stable than the NN. Although the accuracy after 500 iterations by NN is very close to DNN, the stability achieved by train and test in DNN is promising and gives better accuracy in different test datasets. The per iteration accuracy in making success prediction in NN is shown in Figure 4(c) and that in DNN is shown in Figure 4(d). NN is not able to make any improvement after 200 iterations, and the accuracy remains around 86%. But in DNN, test accuracy is around 92%. This demonstrates performance improvement by DNN compared to NN. The accuracy on every iteration for weapon type prediction in NN is shown in Figure 4(e). It can be seen that after 100 iterations, the accuracy remains close to 72%. The accuracy in DNN is shown in Figure 4(f), and after 100 iterations, the accuracy is close to 92%. This demonstrates the improvement in accuracy by DNN compared to NN. The accuracy in NN for making region prediction is shown in Figure 4(g). The maximum accuracy achieved is around 80%. However, more than 95% accuracy is achieved in DNN as shown in Figure 4(h). This experiment demonstrates the performance improvements in DNN as compared to NN. The accuracy in attack type prediction by NN is shown in Figure 4(i). As shown in the figure, the accuracy is around 78%. But the accuracy achieved by DNN is around 92% as shown in Figure 4(j). All these experiments demonstrate that as the number of layers is increased, the network is able to learn more complex nonlinearity in the big data and hence able to make predictions efficiently.

4.4. Comparison of Accuracy, Precision, Recall, and F1-Score in NN and DNN.

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}}, \\ \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{F1} - \text{Score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{7}$$

The formulae to calculate accuracy, precision, recall, and F1-Score are given in equation (7). TP means true positive, TN means true negative, FP means false positive, and FN means false negative. The comparison in accuracy in train and test datasets computed by NN and DNN is given in Figure 5. All these experiments demonstrate that DNN is able to achieve an accuracy of more than 91% in both train and test datasets. The maximum accuracy is achieved in suicide dataset, which is around 98%. The comparison of precision, recall, and F1-Score in test data computed by NN and DNN is given in Figure 6. It can be observed that DNN has achieved more than 91% in precision, recall, and F1-Score. This is another demonstration that as the number of layers is increased, the network is able to learn the features in the dataset and is able to make efficient predictions.

4.5. Confusion Matrix. The confusion matrix is a performance measurement in machine learning classification problems. In case of binary classification, the table is 2 × 2 showing true positive, true negative, false positive, and false negative. In case of multiclass classifications, the table has

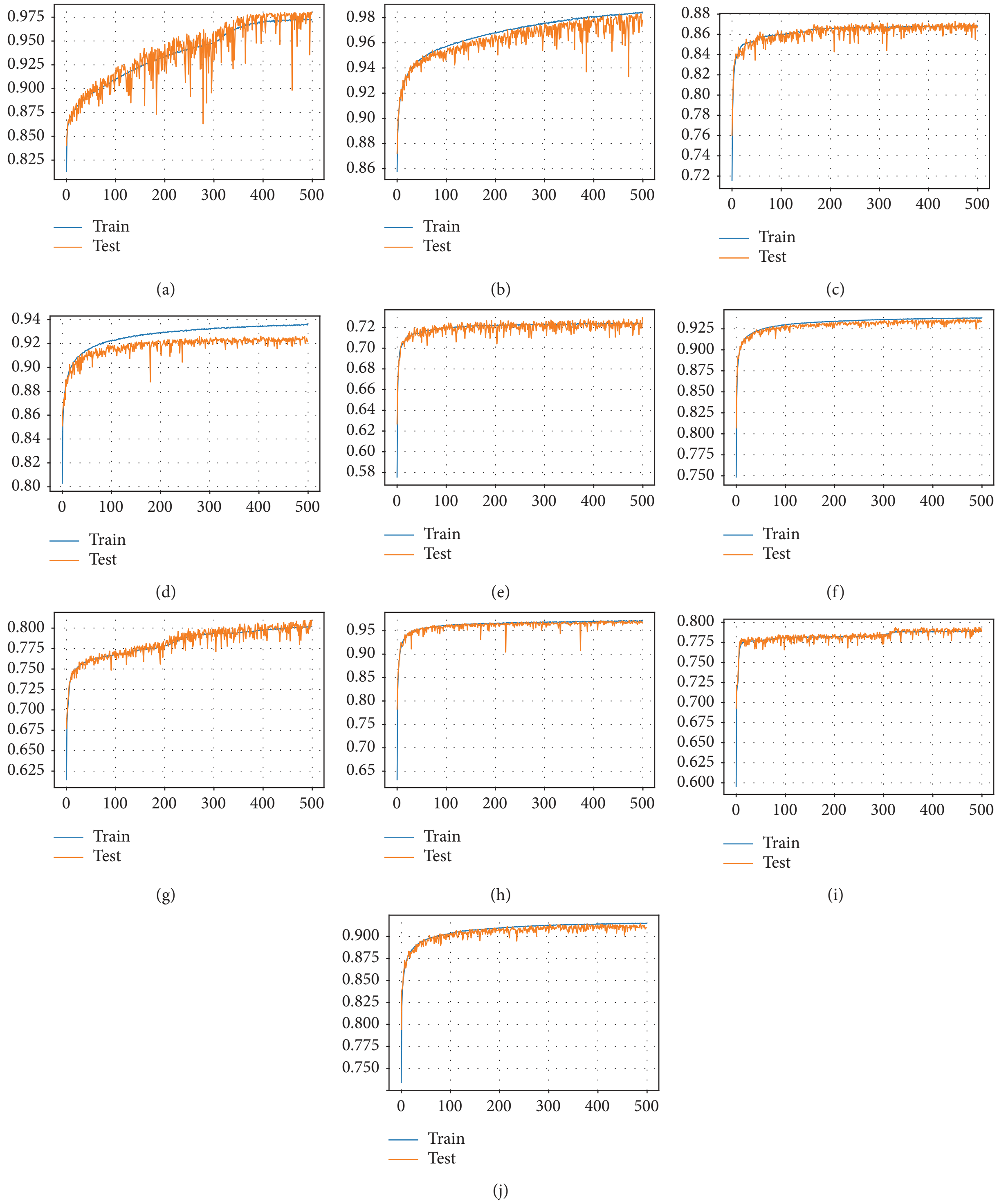


Figure 4: Train and test accuracy in every iteration computed by NN and DNN. (a) Neural network for suicide prediction. (b) Deep neural network for suicide prediction. (c) Neural network for success prediction. (d) Deep neural network for success prediction. (e) Neural network for weapon type prediction. (f) Deep neural network for weapon type prediction. (g) Neural network for region prediction. (h) Deep neural network for region prediction. (i) Neural Network for attack type prediction. (j) Deep neural network for attack type prediction.

size equal to number of classes squared. The confusion matrix computed by DNN for suicide and success is given in Figure 7. The confusion matrix for weapon type, region, and attack type is given in Figure 8. A confusion matrix with large values on the diagonal demonstrate the high accuracy of the model. As shown in these figures, confusion matrix

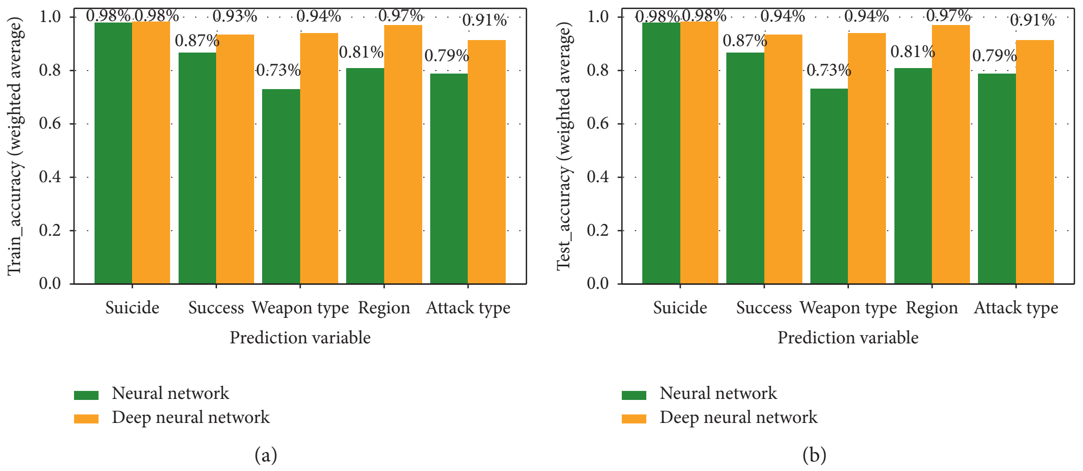


FIGURE 5: Train and test accuracy by neural network and deep neural network in making predictions of suicide, success, weapon type, region, and attack type. (a) Train accuracy by neural network and deep neural network. (b) Test accuracy by neural network and deep neural network.

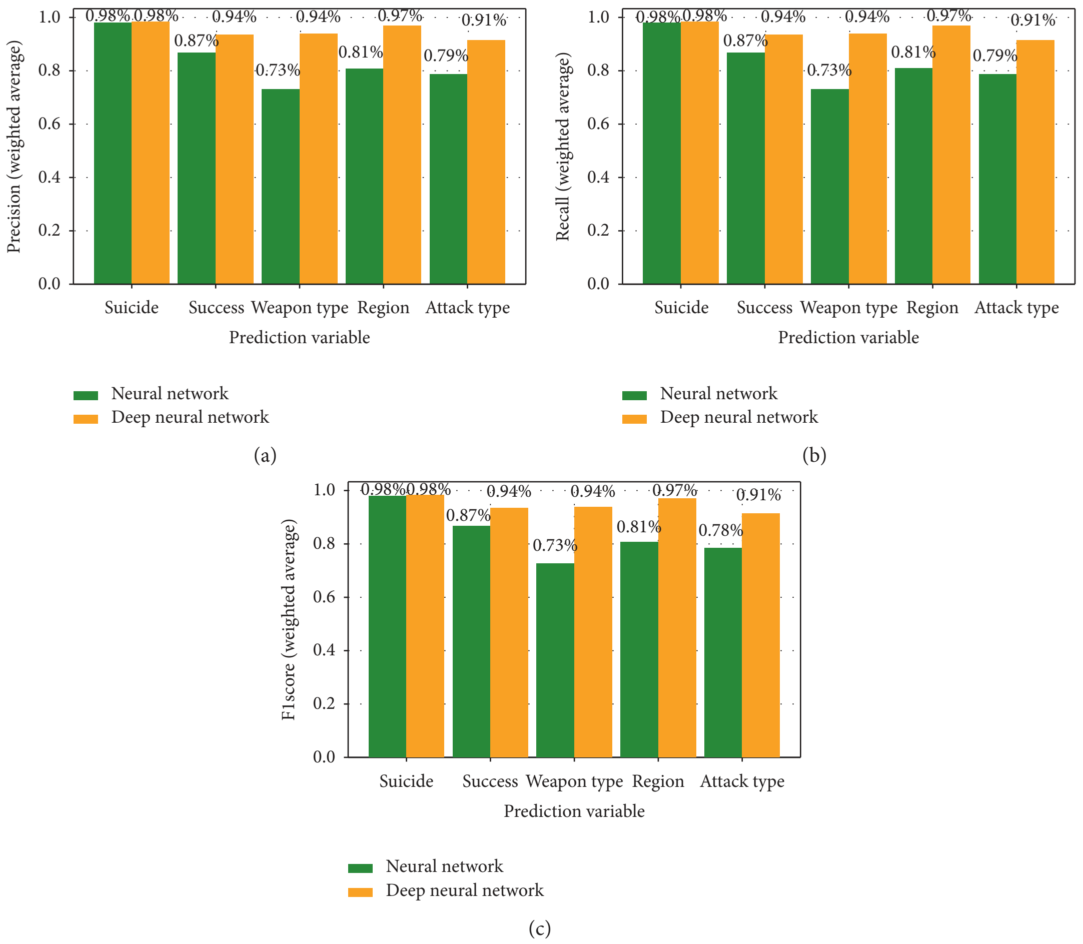
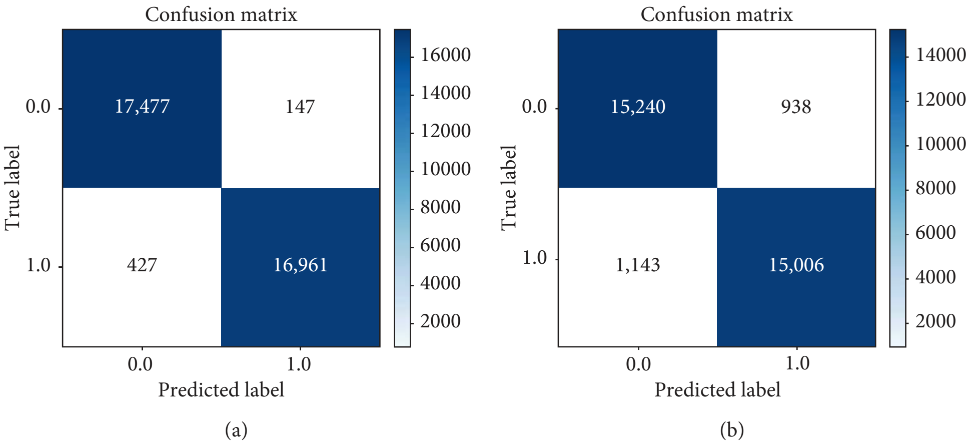


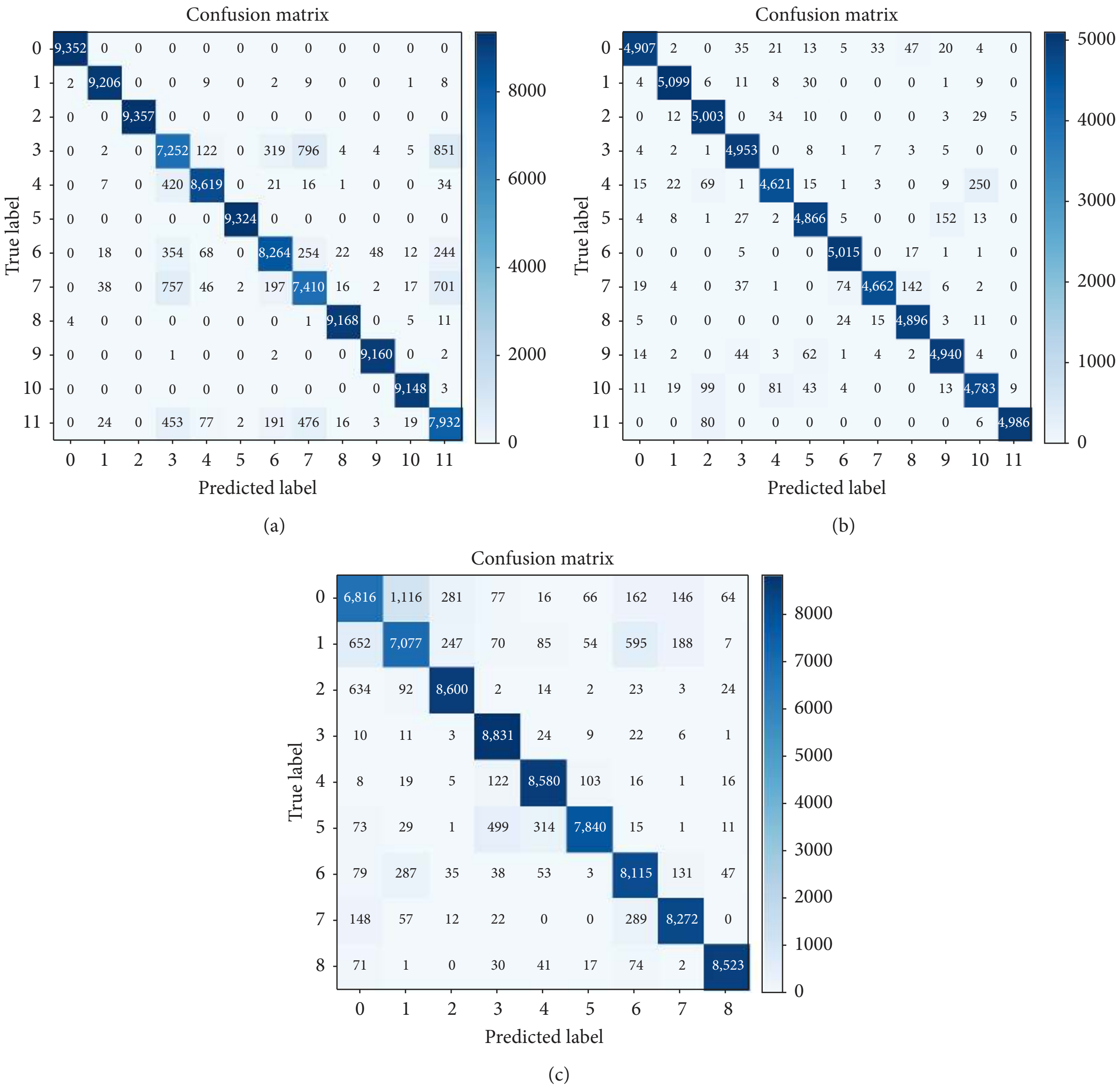
FIGURE 6: Precision, recall, and F1-Score by neural network and deep neural network in making predictions of suicide, success, weapon type, region, and attack type. (a) Precision by neural network and deep neural network. (b) Recall by neural network and deep neural network. (c) F1-Score by neural network and deep neural network.



(a)

(b)

Figure 7: Confusion matrix by deep neural network in making predictions of suicide and success. (a) Confusion matrix of suicide in deep neural network. (b) Confusion matrix of success in deep neural network.

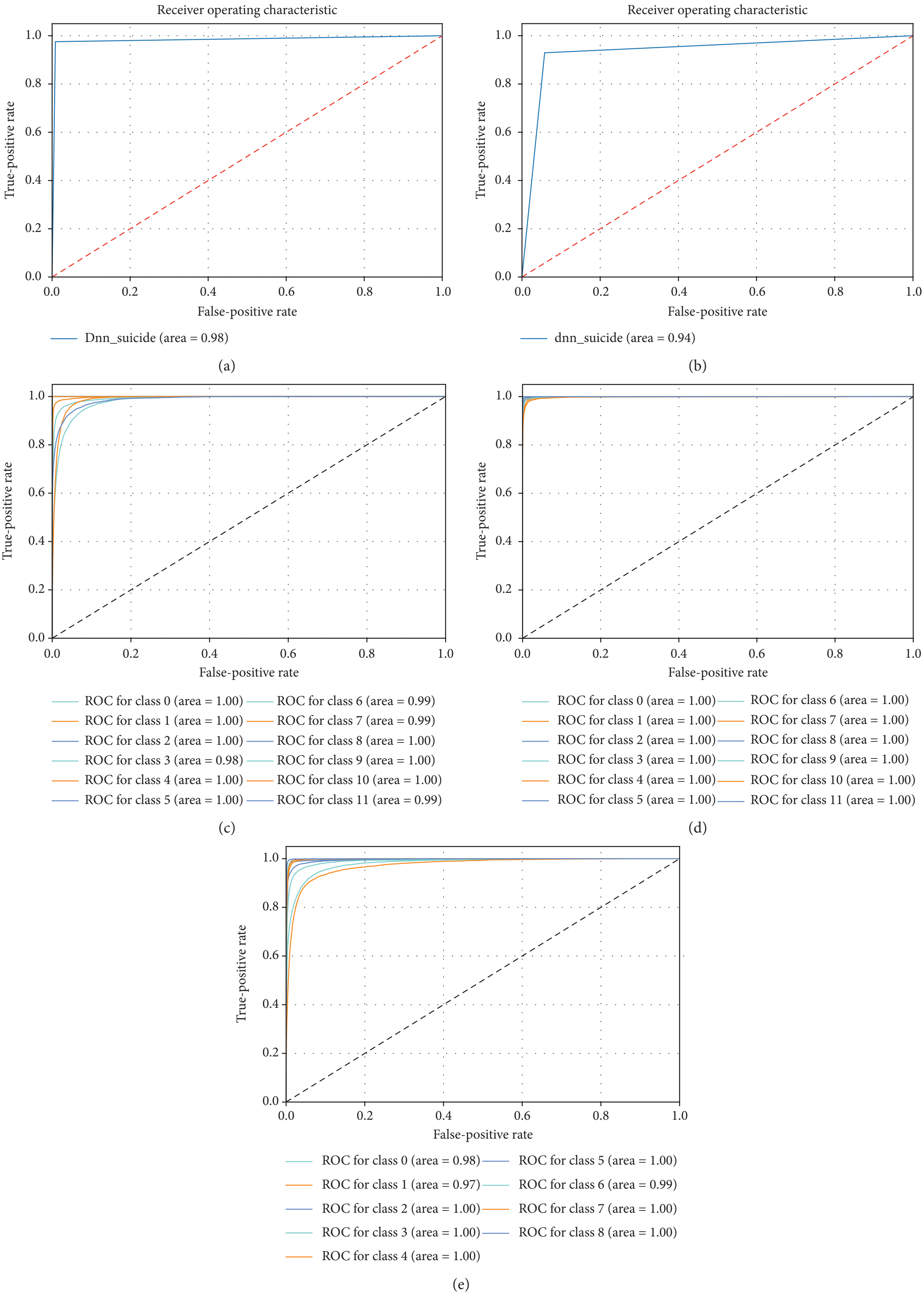


(a)

(b)

(c)

Figure 8: Confusion matrix by deep neural network in making predictions of weapon type, region, and attack type. (a) Confusion matrix of weapon type in deep neural network. (b) Confusion matrix of region in neural network. (c) Confusion matrix of attack type in deep neural network.



(a)

(b)

(c)

(d)

(e)

Figure 9: ROC curve computed by deep neural network in prediction of suicide, success, weapon type, region, and attack type. (a) ROC curve in deep neural network for suicide prediction. (b) ROC curve in deep neural network for success prediction. (c) ROC curve in deep neural network for weapon type prediction. (d) ROC curve in deep neural network for region prediction. (e) ROC curve in deep neural network for attack type prediction.

```
Input: the whole dataset of GTD along with labels
Output: optimized values of $W$ and $b$
Data: GTD Datasets
$W^{[1..L]} = random\_numbers$ //Glorot Uniform initializer
$b^{[1..L]} = random\_numbers$
while $i \leq num\_iteration$ do
    $k \longleftarrow 1$
    while $j \leq L$ do
        $Z^{[j]} = W^{[j]T}.A^{[j-1]} + b^{[j]}$
        $A^{[j]} = g(Z^{[j]})//g(Z) = \max(0, z)$
        increment $j$ by 1
    $\mathscr{L}(A^{[L]}, Y) = -(1/m)\sum_{i=0}^{m} Y_i \log(A_i^{[L]})$ //Binary cross-entropy loss
    $k \longleftarrow L$
    while $k \geq 0$ do
        $W^{[k]} = W^{[k]} - \alpha\partial\mathscr{L}/\partial W[k]$
        $b^{[k]} = b^{[k]} - \alpha\partial\mathscr{L}/\partial b[k]$
        decrement $k$ by 1
```

Algorithm 2: The training of deep neural network using gradient descent optimization algorithm.

Table 2: Working environment.

No. of nodes	Name of machine	Frequency per node (GHz)
48	Intel(R) Xeon(R) Silver 4116 CPU	2.10

Table 3: Performance of comparison of NN and DNN with traditional machine algorithms, i.e., logistic regression, SVM, and Naïve Bayes.

Algorithm	Train Accuracy (%)	Test Accuracy (%)	Average Precision (%)	Average Recall (%)	Average F1-Score (%)
Logistic regression	79.2	76.9	76.7	76.8	76.8
SVM	78.8	78.3	78.2	78.2	78.2
Naïve Bayes	81.3	80.9	80.8	88.8	88.7
NN	83.6	83.6	83.6	83.6	83.6
DNN	94.6	94.8	94.8	94.8	94.8

has high values on the diagonals and hence DNN is proved to be an efficient model for making predictions.

4.6. ROC Curve

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}. \end{aligned} \tag{8}$$

The ROC (receiver operating characteristic) curve shows the performance of the classification model at classification thresholds. The curve shows two parameters: true-positive rate (TPR) and false-positive rate (FPR). These parameters are defined in equation (8). The ROC computed by DNN in making prediction of suicide, success, weapon type, region, and attack type is given in Figure 9. The ROC shows that DNN is able to make classification with accuracy more than 94%.

4.7. Comparison of NN and DNN with Traditional Machine Learning Algorithms. In this section, the performance of the model based on NN and DNN is compared with traditional machine learning algorithms, i.e., logistic regression, SVM, and Naïve Bayes. The comparison in terms of average train and test accuracy and average precision, recall, and F1-Score is shown in Table 3. These results demonstrate that DNN is the most suitable model for this type of dataset as it is an example of big data, where the performance improves when there is big data and a deeper network. Traditional machine learning algorithms such as logistic regression, SVM, and Naïve Bayes including a single-layer NN are not able to capture the pattern in the dataset, and thus the maximum performance of approximately 84% is achieved. But in DNN, it is possible to achieve 95% accuracy on average.

5. Conclusion

Terrorism is the most important threat to the life of mankind of any time. It can affect the quality of life of not only an individual but the whole society. The fear of terrorism restricts people from contributing in the development of the country. In every country, dealing with terrorism is the top most priority of the government. They seek for techniques to understand the different factors involved in terrorism and how to deal with those factors in order to completely stop or reduce terrorist activities. Machine learning is an affective model to learn the different factors of terrorism and can be an important tool for law enforcement agencies to deal with terrorism. In this paper, we have investigated AI-based solutions to understand the different factors of terrorism that can help us make prediction in future terrorist activities. We have identified five different factors that are important to predict for counterterrorism. These factors are whether the type of attack is suicide or not, whether the attack is successful or not, what type of weapon can possibly be used, what region can possibly be targeted, and what type of terrorist is going to be used. We have developed different models based on traditional machine learning techniques, but the results have demonstrated that these models are not able to make predictions with high accuracy. We have developed NN- and DNN-based models, and the results have demonstrated that DNN-based models are the most accurate. The model based on DNN has demonstrated more than 95% accuracy compared to other state-of-the-art techniques in machine learning. These deep learning-based techniques can help governments and law enforcement agencies to understand the factors of terrorism and to design strategies to deal with terrorism before a terrorist activity can actually happen.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant no. DF-461-156-1441. The authors, therefore, gratefully acknowledge the DSR for the technical and financial support.

References

[1] V. Kumar, M. Mazzara, A. Messina, and J. Lee, "A conjoint application of data mining techniques for analysis of global terrorist attacks—prevention and prediction for combating terrorism," *Advances in Intelligent Systems and Computing*, Springer, Berlin, Germany, 2019.

[2] B. Hoffman, "Al qaeda's uncertain future," *Studies in Conflict & Terrorism*, vol. 36, no. 8, pp. 635–653, 2013.

[3] T. Krieger and D. Meierrieks, "What causes terrorism?" *Public Choice*, vol. 147, no. 1-2, pp. 3–27, 2011.

[4] N. Ouassini and A. Verma, "Socio-economic inequality or demographic conditions: a micro-level analysis of terrorism in Jharkhand," *Journal of Victimology and Victim Justice*, vol. 1, no. 1, pp. 63–84, 2018.

[5] R. Alhamdani, M. Abdullah, and I. Sattar, "Recommender system for global terrorist database based on deep learning," *International Journal of Machine Learning and Computing*, vol. 8, pp. 571–576, 2018.

[6] Y. Saeed, K. Ahmed, M. Zareei, A. Zeb, C. Vargas-Rosales, and K. M. Awan, "In-vehicle cognitive route decision using fuzzy modeling and artificial neural network," *IEEE Access*, vol. 7, pp. 20262–20272, 2019.

[7] S. A. A. Shah, I. Uddin, F. Aziz, S. Ahmad, M. A. Al-Khasawneh, and M. Sharaf, "An enhanced deep neural network for predicting workplace absenteeism," *Complexity*, vol. 2020, pp. 1–12, 2020.

[8] I. Uddin, "Multiple levels of abstractions in the simulation of microthreaded many-core architectures," *Open Journal of Modelling and Simulation*, vol. 3, no. 4, pp. 159–190, 2015.

[9] I. Uddin, "High-level simulation of concurrency operations in microthreaded many-core architectures," *GSTF Journal on Computing (JoC)*, vol. 4, p. 21, 2015.

[10] S. Singh, J. Allanach, H. Tu, K. Pattipati, and P. Willett, "Stochastic modeling of a terrorist event via the ASAM system," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 5673–5678, Hague, The Netherlands, October 2004.

[11] A. Torres and C. Tranchita, "Events classification and operation states considering terrorism in security analysis," in *Proceedings of the IEEE PES Power Systems Conference and Exposition*, pp. 1265–1271, New York, NY, USA, October 2004.

[12] R. K. Alex Godwin, R. Chang, and W. Ribarsky, "Visual analysis of entity relationships in the global terrorism database," in *Proceedings of the Defense and Security 2008: Special Sessions on Food Safety, Visual Analytics, Resource Restricted Embedded and Sensor Networks, and 3D Imaging and Display*, vol. 6983, Orlando, FL, USA, 2008.

[13] F. Ozgul, Z. Erdem, and C. Bowerman, "Prediction of unsolved terrorist attacks using group detection algorithms," in *Intelligence and Security Informatics*, H. Chen, C. C. Yang, M. Chau, and S.-H. Li, Eds., pp. 25–30, Springer, Berlin, Germany, 2009.

[14] S. Dixon, M. Dixon, J. Elliott, E. Guest, and D. J. Mullier, "A neural network for counter-terrorism," in *Research and Development in Intelligent Systems XXVIII*, M. Bramer, M. Petridis, and L. Nolle, Eds., pp. 229–234, Springer, London, UK, 2011.

[15] P. H. Pilley, "Review of group prediction model for counter terrorism using clope algorithm," *Journal of Defense Management*, vol. 4, no. 1, 2014.

[16] I. Toure and A. Gangopadhyay, "Real time big data analytics for predicting terrorist incidents," in *Proceedings of the 2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pp. 1–6, Waltham, MA, USA, May 2016.

[17] S. Saha, H. Aladi, A. Kurian, and A. Basu, "Future terrorist attack prediction using machine learning techniques," 2017.

[18] H. Mo, X. Meng, J. Li, and S. Zhao, "Terrorist event prediction based on revealing data," in *Proceedings of the IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pp. 239–244, Beijing, China, March 2017.

[19] F. Ding, Q. Ge, D. Jiang, J. Fu, and M. Hao, "Understanding the dynamics of terrorism events with multiple-discipline datasets and machine learning approach," *PLoS One*, vol. 12, Article ID e0179057, 2017.

[20] P. Garg, H. Garg, and V. Ranga, "Sentiment analysis of the uri terror attack using twitter," in *Proceedings of the 2017 International Conference on Computing, Communication and*

Automation (ICCCA), pp. 17–20, Greater Noida, India, June 2017.

[21] C. Verma, S. Malhotra, and V. Verma, "Predictive modeling of terrorist attacks using machine learning," *International Journal of Pure and Applied Mathematics*, vol. 119, p. 06, 2018.

[22] Z. Li, D. Sun, B. Li, Z. Li, and A. Li, "Terrorist group behavior prediction by wavelet transform-based pattern recognition," *Discrete Dynamics in Nature and Society*, vol. 2018, pp. 1–16, 2018.

[23] X. Zhang, M. Jin, J. Fu, M. Hao, C. Yu, and X. Xie, "On the risk assessment of terrorist attacks coupled with multi-source factors," *ISPRS International Journal of Geo-Information*, vol. 7, no. 9, 2018.

[24] M. Hao, D. Jiang, F. Ding, J. Fu, and S. Chen, "Simulating spatio-temporal patterns of terrorism incidents on the indochina peninsula with GIS and the random forest method," *ISPRS International Journal of Geo-Information*, vol. 8, no. 3, p. 133, 2019.

[25] P. Agarwal, M. Sharma, and S. Chandra, "Comparison of machine learning approaches in the prediction of terrorist attacks," in *Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1–7, Noida, India, August 2019.

[26] S. Kalaiarasi, A. Mehta, D. Bordia, and Sanskar, "Using global terrorism database (GTD) and machine learning algorithms to predict terrorism and threat," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 5995–6000, 2019.

[27] S. P. Maniraj, D. Chaudhary, V. H. Deep, and V. P. Singh, "Data aggregation and terror group prediction using machine learning algorithms," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 1467–1469, 2019.

[28] E. Christie, "*Understanding the dynamics of unclaimed terrorism events in Pakistan: a machine learning approach*," The University of Maine, Orono, ME, USA, 2019.

[29] S. Ahmad, M. Z. Asghar, F. M. Alotaibi, and I. Awan, "Detection and classification of social media-based extremist affiliations using sentiment analysis techniques," *Human-Centric Computing and Information Sciences*, vol. 9, no. 1, 2019.

[30] R. Gui, T. Chen, and H. Nie, "In-depth analysis of railway and company evolution of Yangtze River Delta with deep learning," *Complexity*, vol. 2020, Article ID 5192861, 25 pages, 2020.

[31] X. Yu, Z. Zhang, L. Wu et al., "Deep ensemble learning for human action recognition in still images," *Complexity*, vol. 2020, Article ID 9428612, 23 pages, 2020.

[32] K. Niu, J. Guo, Y. Pan et al., "Multichannel deep attention neural networks for the classification of autism spectrum disorder using neuroimaging and personal characteristic data," *Complexity*, vol. 2020, p. 9, 2020.

[33] A. B. Owen, "Infinitely imbalanced logistic regression," *Journal of Machine Learning Research*, vol. 8, pp. 761–773, 2007.

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

[35] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[36] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[37] K. Gurney, *An Introduction to Neural Networks*, Taylor & Francis, Inc., Bristol, PA, USA, 1997.

[38] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, pp. 807–814, Haifa, Israel, June 2010.

[40] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[41] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, http://arxiv.org/abs/1609.04747.

[42] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[43] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[44] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, Cambridge, MA, USA, 2016.

[46] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, Now Publishers Inc., Hanover, MA, USA, 2014.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2015.

[48] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2, Curran Associates Inc., Lake Tahoe, NV, USA, pp. 2553–2561, 2013.

[49] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[50] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, Sardinia, Italy, May 2010.

[52] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Proceedings of the 4th International Conference on Multiple Classifier Systems, MCS'03*, Springer-Verlag, Cagliari, Italy, pp. 125–134, 2003.

[53] M. Fatima, A. Baig, and I. Uddin, "Reliable and energy efficient mac mechanism for patient monitoring in hospitals," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018.

[54] I. Uddin, A. Baig, and A. A. Minhas, "A controlled environment model for dealing with smart phone addiction," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, 2018.

[55] M. Abadi, P. Barham, J. Chen et al., "Tensorflow: a system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16*, USENIX Association, Savannah, GA, USA, pp. 265–283, November 2016.