# Ensemble Classification for Intrusion Detection via Feature Extraction Based on Deep Learning

**Maryam Yousefnezhad**
sadjad university of technology

**Javad Hamidzadeh** ( ✉ Hamidzadehj@gmail.com )
Sadjad University of Technology    https://orcid.org/0000-0001-6493-0539

**Mohammad Aliannejadi**
university of amsterdam

---

---

# Ensemble Classification for Intrusion Detection via Feature Extraction based on Deep Learning

**Maryam Yousefnezhad** · **Javad Hamidzadeh**⋆ ·
**Mohammad Aliannejadi**

**Abstract** An intrusion detection system is a security system that aims to detect sabotage and intrusions on networks to inform experts of the attack and abuse of the network. Different classification methods have been used in the intrusion detection systems such as fuzzy, genetic algorithms, decision trees, artificial neural networks, and support vector machines. Moreover, ensemble classifiers have shown more robust and effective performance for various tasks in the field. In this paper, we adopt ensemble models in order to improve the performance of intrusion detection and, at the same time, decrease the false alarm rate. We use kNN for multi-class classification, as well as SVM to approach the classification problem in normal-based detection. In order to combine multiple outputs, we use the Dempster-Shafer method in which there is the possibility of explicit retrieval of uncertainty. Moreover, we utilize deep learning for extracting features to train the samples, selected by the sample selection algorithm based on ensemble margin. We compare our results with state-of-the-art methods on benchmarking datasets such as UNSW-NB15, CICIDS2017, and NSL-KDD. Our proposed method indicate the superiority in terms of prominent metrics Accuracy, Precision, Recall, and F-measure.

**Keywords** Intrusion detection system · Dempster Shafer · Deep learning · Ensemble margin · Classification

⋆ Corresponding author.

F. Maryam Yousefnezhad
Faculty of Computer Engineering and Information Technology,Sadjad University of Technology,Mashhad
E-mail: m.yousefnejad112@sadjad.ac.ir

S. Javad Hamidzadeh
Faculty of Computer Engineering and Information Technology,Sadjad University of Technology,Mashhad
E-mail: j_hamidzadeh@sadjad.ac.ir

Mohammad Aliannejadi
IRLab, University of Amsterdam, Amsterdam, The Netherlands E-mail: m.aliannejadi@uva.nl
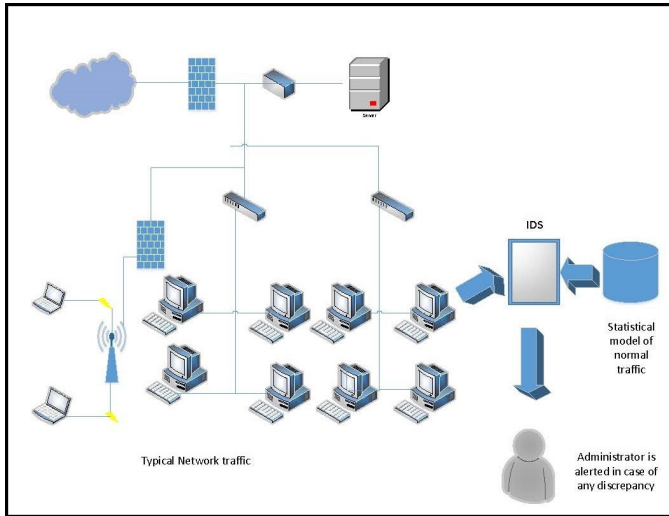
**Fig. 1** Anomaly-based detection method

## 1 Introduction

Nowadays, most security systems are mainly focused on encoding, firewall, and access control [48]. However, these methods mainly suffer from network security and cannot guarantee system and network security. Intrusion Detection Systems (IDS) aim to improve systems' security, playing an important role in Cyber Security [33]. Intrusion is defined as a set of hypothetical operations for security purposes, integrity, confidentiality, or availability. Events that enter to network or host are monitored by the intrusion detection system, and necessary measures are taken in relation to whether these events are a sign of an attack or proper use of the system [35, 50]. Generally, intrusion detection methods include: (i) Anomaly-based (ii) Signature-based. In the following, we describe both categories of methods.

– **Anomaly-based detection method:** In this method, normal patterns and behavior should be first identified, and special patterns and rules should be found for them. Behaviors that follow this pattern are known as normal behavior, and behaviors that are significantly deviated from these patterns are considered to be anomaly behaviors [2, 10]. An anomaly-based intrusion detection system is shown in Figure. 1.
– **Signature-based intrusion method:** In this method, pre-determined attacks and intrusions patterns are kept as a rule inside the database, and each pattern represents an intrusion. In this method, Network traffic is examined, and the occurrence of intrusion is announced if there is such a pattern in the system [2]. a signature-based intrusion detection system is shown in Figure 2.

Accuracy and precise detection are of the most importance in the intrusion detection system. In this regard, we are providing an intrusion detection system that can
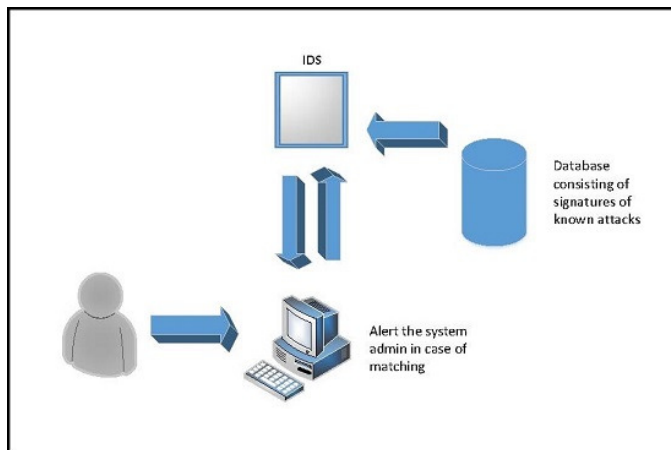
**Fig. 2** Signature-based intrusion detection system

classify attacks. This system, with a higher degree of accuracy, has a significant impact on system performance, and machine learning is a reliable tool in this field [28].

According to the results of various experiences about the use of training methods, there is no single specific training algorithm that works better and more effectively for all applications [52]. In fact, each algorithm is a specific model formed based on certain assumptions. Sometimes these assumptions are true, and sometimes these assumptions are violated. Therefore, no algorithm alone can work successfully in all conditions and for all. Ensemble methods have been introduced in order to overcome this problem [52, 36]. Ensemble methods have been used in classification in the recent decade [31, 21]. These methods have better performance than single methods. In this study, we propose evidence theory is a mathematical theory based on the posterior probabilities to combine the evidence from kNN and SVM classifiers such as the final decision-making for improving the ability and increasing accuracy of intrusion detection. The contributions of the article can be summarized as follows:

1. Using ensemble margin for better sample selection;
2. Using deep learning for feature extraction;
3. Using ensemble method Dempster–Shafer for combine classifier;
4. Conducting extensive experiments to evaluate the performance of the proposed method on KDD-Cup and NSL-KDD data sets.

In order to ensure diversity, we trained four support vector machine classifiers and four probability K-nearest neighbor algorithms. Hermit function used for reducing the number of support vectors and enhancing the accuracy of data classification and Reduction of support vectors could lead to improving the speed of data classification [32]. We used the Sigmoid function to make the probability support vector machine, and deep learning was used to extract key quality features from among 41 features in the KDD99 data set, sample selection algorithm is a crucial task in sample-based learning algorithms. We used that to select better samples based on the

ensemble margin [17]. Finally, we used from Dempster-Shafer evidence theory for fusion data. This theory strengthens correct decisions and weak incorrect decisions based on probability. We perform an extensive set of experiments in which we show that our proposed method can outperform state-of-the-art approaches for detecting intrusion attacks.

The rest of this paper is structured as follows. In section 2, we review the literature in the field of intrusion detection. We explain the background in section 3. Section 4 describes our methodology, followed by section 5 explaining the experiments on the data sets. Finally, the conclusion and the suggestions for future work are presented in section 6.

## 2 Literature review

In the 1970s, the need for security systems is felt more than ever due to the increasing speed, efficiency, number of computers. In 1977 and 1978, the International Standard Organization held a meeting between governments and inspection bodies of Electronic Data Processing that the outcome that meeting was to prepare a report on the status of security, inspection, and control of systems at that time. At the same time, the US Department of Energy began very detailed studies on the inspection and the security of computer systems due to concerns about the security of its systems. This study was carried out by a person named James P. Anderson. The Report presented by Anderson in 1980, can be introduced as the main core of the concepts of intrusion detection [19]. Raman Singh et al. [42] presented a proposed system based on Extreme Learning Machine. This machine solves the problem of the neural network in terms of speed. This system was used aimed to reduce computational memory and time using creating a profile of network traffic, and as well as two alpha and beta profiles were used. Alpha and beta profiles can reduce the effect of unaligned data. The beta profile can reduce the size of the experimental data set, while its features are maintained in practice, and the alpha profile is used to reduce the effect of discovery time. Folino et al. [11] used an intrusion detection system based on ensemble classification, aimed to increase group accuracy. The ensemble structure of the NIDS makes possible the detection of sophisticated attacks and alarms in a proper manner, and the advantages of using this ensemble classifier include reducing error variance and bias, and it is appropriate for unbalanced classification. The proposed method works well to identify attacks and minimize the alarms but needs to be improved for specific attacks. Aburomman et al. [1] presented a new method based on the support vector machine, K-Nearest Neighbor, and particle algorithm, and the weighted majority algorithm classifier for the intrusion detection system. Six support vector machine classifiers and six K-Nearest Neighbor classifiers with different values have been used in this method. Then, WMA was used as a classifier combination. The Local uni-modal sampling (LUS) algorithm was used to select high-quality parameters. The proposed method has used LUS-WMA that has better accuracy than a method that uses the WMA classifier, but the performance of WMA alone is better than the proposed method. Gautam et al. [14] used the proposed algorithm based on information theory and entropy, in which this algorithm obtained the entropy after the classi-

fication of features, and classification is based on bias and features. The results show that the rate of detection and accuracy of the proposed algorithm is better than the Fast Feature Reduction in Intrusion Detection Data sets (FFRIDD) and Multi-Level Dimensionality Reduction Methods (MLDRM) selection algorithm. A hybrid semi-supervised learning technique was introduced using the Active smart vector learning machine (ASVM) and Fuzzy C-Means (FCM) in the design of an intrusion detection system that has an excellent performance. This system is considered as a binary classification and hence works faster than multi-classifiers [24]. Li et al. [27] presented a new hybrid method based on the density peaks clustering and k nearest neighbors in order to increase the accuracy rate that DPNN was used to train, and kNN was used for classification. Finally, the proposed DPNN method has better accuracy than the support vector machine, and there are many other methods in the field of machine learning. Vinayakumar et al. presented a proposed hybrid intrusion detection system (Scale-Hybrid-IDS-Alert Network) based on a high level of a scalable framework on a hardware server that the capability to classify unpredictable cyber-attacks, monitor network, and host-level event. The framework distributed based on a deep learning model with the DNNs method used for analyzing big data in real-time and optimal network parameters and network typologies for DNNs. Based on the tests obtained, the performance of the DNNs is higher than that of the classical method [46]. El-Sappagh et al. [39] depicted different classification methods of data mining for true detection false alarm and high accuracy. Based on many methods of data mining on KKD CUP99 that disclose all attack classes with high accuracy. In this paper, the best accuracy gain in the multilayer perceptron method by 92%, and the best training time in the rule-based model is 4 seconds. Elmasry et al. [9] proposed a method using an ensemble weighted majority algorithm to increase accuracy a feature selection method to decrease the number of features for detection attacks. This method increase accuracy of detection by 10, false-positive rate, reduces to 0.05%. Zhang et al. [51] proposed class imbalance processing technology for IDS data set, which combines Synthetic Minority Over-Sampling Technique (SMOTE) and under-sampling for clustering based on Gaussian Mixture Model (GMM). The advantage of their novel method verified using the UNSW-NB15 and CICIDS2017 data sets. This model shows an effective solution to imbalanced data in an intrusion detection system.

Given the literature mentioned above, most of the methods employed have focused on increasing accuracy and precision and redaction false alerts. In the present paper, we used an ensemble method in order to increase the accuracy rate of intrusion detection system for the classification of multiple class attacks. We also used Deep learning in order to reduce time and select better features.

## 3 Background

In Section 3.1 we discuss Dempster-Shafer theory of evidence and parameters. Feature extraction using Autoencoder illustrated in section3.2

3.1 Dempster–Shafer (DS) theory

Evidence Theory (ET) was first presented by the publication of the theory of upper and lower probabilities in 1976 [12]. Shafer developed this theory and fixed its deficiencies. It is used as a tool to analyze uncertainty in the theory of inaccurate probabilities [49], which later became known as the theory of belief functions. The Dempster–Shafer theory is important by discussing existing beliefs about a situation. This theory is considered the most effective method in integrating data at the general level [16]. Basic Functions in the Dempster–Shafer Theory include:

1. The probability mass function
2. The Belief function
3. The Plausibility Function

Here, we consider a finite set $\phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$ that called the frame of discernment (FOD), which includes all subsets of $\phi$. Function mass is defined m : $2^\phi \longrightarrow [0, 1]$ where $\phi$ is named the power set and illustrate by $2^\phi$.

The probability mass function, known as the mass function, is considered as the essential function and known with the two symbols m and basic probability assignments (BPA). The probability mass function is an evidence for the existence of state A, which is defined as a real number between zero and one, and the probability of occurrence $\phi$ is zero, and the sum of the subsets is one.

$$m : 2^\phi \longrightarrow [0, 1] \qquad if \qquad m(\phi) = 0 \tag{1}$$

$$\sum_{A \varepsilon \phi} m(A) = 1 \tag{2}$$

Belief function means the lowest probability limit a state that may occur, and the Plausibility function is the upper limit of the belief in reality. A function Bel: $2^\phi \longrightarrow [0, 1]$ is named belief function over the frame of discernment$\phi$ . A function Pl: $2^\phi \longrightarrow [0, 1]$is named the Plausibility function one the frame of discernment(FOD) $\phi$ . The Plausibility function by a function is called doubt, is related to the belief function and According to the following relation is defined as a belief.

$$Bel(A) = \sum_{B|B \subseteq A} m(B) \tag{3}$$

$$pl(A) = \sum_{B|B \cap A \neq \phi} m(B) \tag{4}$$

Belief interval reflects the uncertainty that The probability of the occurrence of A shown by P(A)

$$Bel(A) \leqslant p(A) \leqslant pl(A) \tag{5}$$

The Dempster's rule of combination: in here Suppose $m_1$ and $m_2$ are basic probability assignments with occurrence element $B_1, B_2, ..., B_i$ and $C_1, C_2, ..., C_j$,respectively According to the assumptions by The equation 6

$$\sum_{B \cap C = A} m_1(B) * m_2(C) \leq 1 \tag{6}$$

Dempster's rule of combination to combine the output of classes is presented in equations by Equations 7,8,9,10.

$$M_{12}(A) = m_1(B) + m_2(C) = \frac{\sum_{B \cap C = A} m_1(B) * m_2(C)}{1 - K} \qquad When \qquad A \neq \phi \tag{7}$$

$$K = \sum_{B \cap C = \phi} m_1 B * m_2(c) \tag{8}$$

$$m_1(A) + m_2(A) = (1 - K^{-1}) * \sum_{\cap A_i = A} m_i(A_i) \tag{9}$$

$$K = \sum_{\cap A_i = \phi} m_i(A_i) \tag{10}$$

### 3.2 Feature extraction using deep learning

A deep neural network is considered as a general modification for a set of multi-layered architect neural networks that show how a neural network with a large number of layers can work successfully in creating the necessary structures for deep learning. Nowadays, deep learning artificial neural networks have had a lot of competition in pattern and machine learning detection. Feature learning algorithms are used for finding and extracting of common patterns automatically in order to use the extracted features in regression and categorization processes [41]. Also, they are considered as a new method for artificial neural networks, which use large-scale and cost-effective computing. These methods are used to recognize a visual face, dimension detection, network intrusion detection, and many other domains [26].

This learning technique can not only be used to generate significant indicators for the data set due to the correct architecture, reducing the number of neurons in the layer but also to compress it to create compact features, reduces the number of intermediate-layer neurons, which is called an autoencoder.

In the simplest form, an autoencoder has three layers in which there is an input layer and an output layer, and a hidden layer. If there is a p neuron in the input and output layer and a q neuron in the hidden layer, and $F : R \rightarrow R$ is a transfer function, such as the sigmoid function, $X \varepsilon R_p$ is an input vector of the features [15]. The value of the features in a neuron of the hidden layer is obtained using the following equation.

$$H_i(X) := F(W_i^T(X) + b_i) \quad for \qquad all \quad i = 1, \ldots, q \tag{11}$$

Where $W_i \varepsilon R_p$ and $b_i \varepsilon R$, are the corresponding weight and bias parameters of the neuron i, respectively.

After computing the features in the hidden layer as follows:

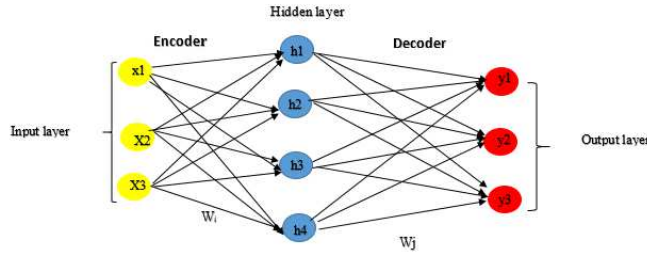$$H := [h_1(x), h_2(x), \ldots, h_q(x)] T \varepsilon R_q \tag{12}$$

**Fig. 3** Auto encoder network.

The representation created in this layer is used as input of the output layer and the output of the last layer of the neural network for all values:

$$Y_j(h) := F'(W_j^T(h) + b_j) \quad \text{for} \quad \text{all} \quad j = 1, \dots, p \tag{13}$$

Where $W_j \varepsilon R_p$ and $b_j \varepsilon R$ is the corresponding weight and bias parameters to j neurons in the output layer.

$$W := [W_1, W_2, W_3, \dots, W_q]^i \varepsilon \quad R^{p*q} \tag{14}$$

$$W := W' = [W_1', W_2', \dots, W'^T] \varepsilon \quad R^{p*q} \tag{15}$$

Weights matrices are found in the entire network. And the transfer functions $F'$, F are not the same necessarily [26]. For inputs, the most common cost function is the mean-square error [47].

$$L(X, Y, W, W') = \frac{1}{2}\|X - Y\|_2{}^F + \frac{\lambda_1}{2}\|\|X - Y\|_2^F + \frac{\lambda_2}{2}\|X - Y\|_2^F \tag{16}$$

Where, the cost function is composed of two distinct parts. An autoencoder network is shown in Figure. 3.

The above structure is expanded in order to extend the auto-encoder method in deep learning methods, and the result of the hidden layer in the first auto-encoder is used as the input of the second auto-encoder, this process can continue for increasing the deep neural network layers. The method described above was used in the training of the final network so that the first hidden layer is trained using the method described and are stabilized after creating the weights of the hidden layer inputs and the next hidden convergence is trained like an auto-encoder in a similar method, and this process continues until the entire network architecture is completed. Finally, the setup step is done on the entire network structure in an integrated manner. Finally, after a training network, the encoder output is considered as the extracted feature.

## 4 Methodology

In this paper, as stated before, we present an intrusion detection system using belief function theory to achieve a higher accuracy rate in the classification and detection of attacks. The block diagram of the proposed method is given in Figure. 4. Therefore, the framework of the proposed system is provided in section 4.1.Ensemble margin is introduced in section **??** to removing the noise and redundancy. kNN probability and its parameters are presented in section 4.4, and finally, a method of probability SVM is provided in section 4.5. Now, we will explain the proposed method in detail.

4.1 Framework of the proposed system

In the framework of the proposed system, the aim is to develop ensemble-based classifiers that will enhance the accuracy of classification attacks in the intrusion detection system. For this purpose, we trained and tested eight classifiers in which the results of four kNN classifiers with the nearest neighbor values of k=3, k=5, k=8, k=10, and four SVM classifiers with RBF kernel and the values of RBF=1, RBF=3, and hermit kernel with degree=8, degree=10 are combined in one group using combine classification. Here, we used four experts of SVM and kNN for diversity, and this leads to a higher performance of the proposed system.

In order to integrate the data, we used combining ensemble classification of Dempster-Shafer due to the possibility of explicit retrieval of uncertainty. This method can integrate numerical, signal, and multidimensional data and is considered the most powerful method of data integration. We used the heuristic function in kNN for converting the format of its output to the probable output. Also, the sigmoid function is used for converting SVM output to probabilistic values. Sigmoid function has a better performance than linear and polynomial functions in SVM experts. Probabilitizing the outputs of the classifiers used in the proposed method is done because Dempster-Shafer's theory is based on probabilities and uncertainty, and its input should be as probable values. In Figure 4, the framework of the proposed method is divided into nine stages.

1. Data pre-processing
2. Sample selection using ensemble margin
3. Features Extract data set using deep learning
4. Data classification with four SVM with RBF and Hermit kernels
5. Make the output of SVM probabilistic using the sigmoid function
6. Make kNN probabilistic using heuristic function
7. Data classification with four probabilistic kNN with a different value of
8. Using the Dempster-Shafer Rule to get the final values BPA
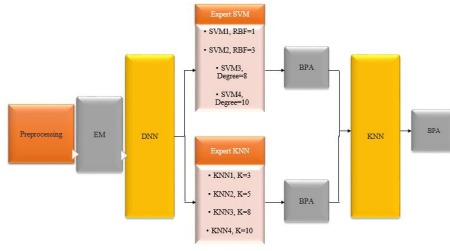9. Analyze and determine the final classification results

**Fig. 4** System framework

### 4.2 Ensemble Margin

The ensemble margin for the first time was presented by Schapiro [40]. In order to explain the success of the boosting algorithm. The ensemble margin is considered as an important concept in ensemble learning that has a very good accuracy rate of ensemble learning, and its values vary between zero to one. The ensemble margin can be calculated in the form of the difference between votes for samples in the feature space $x_i$ where $c1$ is the class that has the most votes among other classes and $m_{c_1}$ is the number of votes, and $c_2$ is the second class with the most votes and $m_{c_2}$ is the number of its votes, and L is the number of classifiers [38].

$$Margin = M = \frac{mc1 - mc2}{L} \tag{17}$$

The mean value of the ensemble margin is obtained from the following equation.

$$Average\ square\ margin = \frac{1}{N} \sum_{i=1}^{10} M^2 \tag{18}$$

### 4.3 Sample selection algorithm based on ensemble margin

In the proposed method, we used the sample selection algorithm to select the experimental and experimental samples from the KDD99 data set based on the Ensemble Margin (EM), which is a fundamental concept in ensemble learning. The steps are as follows:

1. Giving the selected data set, initializing for the number of classes and classifications used in the problem, and Initializing the threshold number $\gamma$, $\gamma = 0.633$.
2. Implementation of the proposed algorithm and calculating the ensemble margin for each educational sample.
3. Selecting better samples using threshold $\gamma$.
4. item Putting the selected experimental sample in the data set $S'$

We used the new data set in the proposed algorithm after selecting the best samples using the feature selection algorithm based on the ensemble margin. The pseudo-code of the proposed algorithm is shown below.

---

**Algorithm 1** Sample Selection Algorithm

---

**Input:**
  $S$ : KDD cup 99 Data set
  $L$: Number of Classifiers
  $C$: Number of Classes
**Output:**  Reduce data set $S$ to $S'$
  **while** Execute Purpose Method **do**
      **for all** $i \in S$ **do**
          **for all** $j \in L$ **do**
              Compute the Ensemble Margin with define by Eq.18 for data set $S$
          **end for**
      **end for**
      **for all** $i \in S$ **do**
          **if** $EnsembleMargin(X_i) < \gamma$ **then**
              $S' \leftarrow X_i$
          **end if**
      **end for**
  **end while**

---



**Fig. 5** k-nearest neighbor

## 4.4 Making probability of the nearest neighbor algorithm

K-nearest neighbor classifiers (kNN) are one of the classification algorithms that widespread used for supervised learning tasks; it is a practical algorithm for classification, and most popular is due to the simplicity of the concept and rapid implementation. This classifier was introduced by P.E.Hart and T.M.Cover [7]. We used The kNN classifier for solving multi-class problems. In the present study, four probabilistic kNN classifiers with different k values are used for classification attacks.

The different values of k are considered for the diversity and better performance of the classifier. The probabilistic kNN method is based on the posterior probability estimate, and we used the Heuristic function to convert the classifier output to probable values. The kNN proposed algorithm is shown below.

---

**Algorithm 2** PKNN Algorithm

---

**Input:**
  $X$: Input data
  $K$:K Number of nearest neighbor
  $N$: Number of data
  $m$: Number of classes
**Output:** $Pi$
    $sum = 0$
    $d_{(x,y)} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$
    **for all** $L \in K_n$ **do**
        find k nearest to x        $k_1, k_2, ..., k_m$
    **end for**
    **for all** $i \in k_m$ **do**
        **for all** $j \in k_i$ **do**
            $sum_i = sum + (d_j)$
            $sum = sum_i$
        **end for**
        **if** $sum_i = 0$ **then**
            $p(i) = 0$
        **else**
            $P(i) = \frac{\sum_{i=1}^{k}(d_i)}{sum}$
        **end if**
        $p(x_i) = \max(P(i))$

---

We assumed the probabilistic method in such a way that there are a few points with different features in space. Three different classes are shown in Figure 5. In this method, the Euclidean distance of the K nearest neighbor from the Ex experiment data obtained using Eq.19.

$$\mathbf{d_{(x,y)}} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \tag{19}$$

The steps to obtain Probabilistic values kNN are as follows:

1. The first step we computed Euclidean distance for the N input data;
2. The second step we obtained k nearest neighbor from the Ex experiment data;
3. The third step we obtained the distance of m date related to class $i$ from the Ex data, where $d_m$ is the euclidean distance of the m-th data from the test data;
4. The fourth step we computed the probability $p_i$ $i = 1...m$ using Eq.21;
5. The sixth step we opted Maximum amount $p_i$ that data with Possibility $p_i$ belonging to class i.

$$Distance = \sum_{i=1}^{m} d_m(x, y) = d_1 + d_2 + ... + d_m \tag{20}$$

$$p(i) = \frac{\sum_{i=1}^{k} d_i(x, y)}{Distance} \qquad i = 1, ..., k \tag{21}$$

As shown in Figure 3 which is $k = 5$ (the nearest neighbor value),in this figure two data belong the class i and two data belong the $i + 1$ class, and one data belongs the $i + 2$ class. To obtain the probabilistic value of the output of kNN.

First, we have to compute the probability of all classes. In order to compute the probability of belonging test data to class I, we must obtain the first data interval, and the second data belonging to class I using Euclidean distance, then we calculate the total inverse distance of these two test data. Then we gain the sum of Euclidean distance of the six test data. Finally, the probability of belonging to the class I calculate using Eq. 21. For all classes, we do the above stage. After calculating the probabilistic values of the data classes, if the value of p(i) is greater than p(i+1) and p(i+2), the experimental data with the probability p(i) belongs to class i.

4.5 Probabilistic support vector machine (PSVM)

Support vector machine(SVM) is one of the best methods for solving multi-class and regression problems. The support vector machine can be used well in two-class to multi-class classification problems. It has a high training rate and decision-making speed and is appropriate for solving regression and classification problems. Among the two methods of One-against-all(OAA) and One-against-another, that is used to generalize vector support machines to multi-class mode; the One-against-all method is used for implementation [18]. Different kernels, such as linear, polynomial, sigmoid, RBF are introduced for use in the feature space .

Experimental results have shown that The SVM classifier has better performance in classification using RBF kernel and hermit. Hermit kernel using for increases the classification accuracy and the classification speed.

Liner Kernel Function:

$$k(x, y) = x^T y \tag{22}$$

Polynomial Kernel Function:

$$k(x, y) = [(x^T y) + c]^p, p \in N, c \geq 0 \tag{23}$$

Sigmoid Kernels Function (SKF):

$$k(x, y) = \tanh(a(x, y) + v), a \geq 0, v \geq 0 \tag{24}$$

Radial Basis Function (RBF):

$$K(x, y) = \exp(-\delta |x - y|^2), \delta > 0 \tag{25}$$

Hermit Kernels Function:

$$k(x, z) = \prod_{j=1}^{d} \sum_{i=1}^{n} He_i(x_i) * He_i(z_j) \tag{26}$$

$$He_0(x) = 1, He_1(x) = x, He_n(x) = xHe_{n-1}(x) - (n-1)He_{n-2}(x) \tag{27}$$

In this paper, we used different kernels for the experiment. The experimental results illustrated the best result obtained with the RBF kernel function. We train four different expert SVM with RBF kernel functions. This approach used for more diversity of experts in ensemble classifiers. we selected values for RBF kernel function
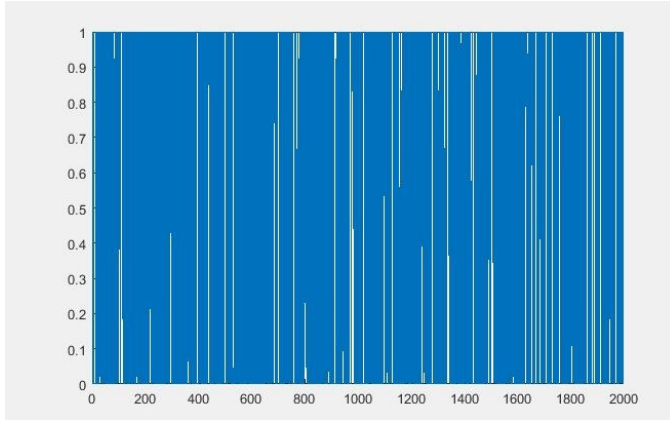
**Fig. 6** probabilistic output SVM

that defined in vector with value=[1,3 ] and hermit kernel function with degree=[8,10 ].

SVM output is a distance used for the comparison of the classifiers. However, posterior possibilities are needed for most applications. Platt introduced a method for converting SVM output to probabilistic values using a sigmoid function to obtain a maximum similarity [37], and this function is based on the Levenberg-Marquardt algorithm and defined by Eq.28, which is a belief model algorithm.

There are linear methods and polynomial functions for the probability of a backup vector machine. Still, the use of the sigmoid function has better results according to the results of the experiments. As shown in Figure 6 the probabilistic output for two thousand test samples. In the present study, the sigmoid function used for obtaining the probability of the output of SVM.

$$P_i = \frac{1}{1 + \exp(Af_i + B)} \tag{28}$$

We showed the algorithm for sigmoid Training, and this function is the model-trust algorithm, based on the Levenberg-Marquardt algorithm. PSVM Algorithm shown below. In this algorithm vector of outputs of the SVM on a data set, data set labels, Number of Negative points, and Number of positive points using as input data, and the coefficients A and B such that the posterior probability of $P(y = 1|x)$ is output algorithm, where f(x) is the output of the SVM.

## 5 Experiments

### 5.1 Data set KDD

The network traffic data set is used to evaluate the proposed method. The Cyber systems and MIT Lincoln Laboratory Technology Group have collected a network

---

**Algorithm 3** PSVM Algorithm

---

**Input:**
   Vector of outputs of the SVM
   Labels
   *Prior*0: Number of Negative points
   *prior*1: Number of positive points
   *Htarget*= (prior1+1)(prior1+2)
   *Ltarget*= 1/(prior0+2)
**Output:** *A* and *B* and *P_i*
      A=0
      B=log(prior0+1)(prior1+)
      **while** minimum function $(ti - 1)log(1 - pi) + (tilog(pi))$ **do**
         update and calculate gradient and hessian of error function(use H'=H+Sigma I
         compute *Pi* define by Eq.28
      **end while**
      find the coefficients A and B such that the posterior probability *P_i*

---

**Table 1** Number of observations in KDD99

| Connection type | Train data set | Test data set |
|---|---|---|
| Normal | 97298 | 60593 |
| Attack | 396743 | 250436 |
| Total | 494021 | 311029 |

traffic data set. This lab simulates the United States Air Force (USAF) LAN network with 9-week continuous attacks of TCP-dump data. This data set by simulation of various attacks such as normal traffic and four classes of attacks [20, 44]:

– DOS: in this attack, the attacker sends a large number of requests to a host.
– R2L: the attacker is trying to unauthorized access from an external machine to the root of the system.
– U2R: the attacker attempts to use vulnerable points of the system to control an external machine through the network as a local user.
– Probe: the attacker is trying to get information about machines and network services.

This work has been done on various platforms such as Windows and Unix. The KDD99 has five classes that are listed in Table 4. The number of educational and experimental records is shown below.

The NSL-KDD is a refined version of the KDD99. In this data set redundant, duplicate records are eliminated for the decline biases of the classifier. It consisted of all the featured needed to form the KDD.The KDD data set as the benchmark data set as well as the most popular database in the field of detection and intrusion has 41 features with 38 numerical features and 3 symbolic features.

## 5.2 Data set UNSW-NB15

The cybersecurity research team of the Australian Centre for Cyber Security (ACCS) has introduced a new data called UNSW-NB15 [45] to resolve the issues found in

**Table 2** UNSW-NB15 Instances

| Attack types | UNSW-NB15 | UNSW-NB15-TRAIN-1 | UNSW-NB15-VAL | UNSW-NB15-TEST |
|---|---|---|---|---|
| NORMAL | 56,000 | 241,911 | 14,089 | 37,000 |
| Generic | 40,000 | 30,081 | 9919 | 18,871 |
| Exploits | 33,393 | 25,034 | 8359 | 11,132 |
| Fuzzers | b18,184 | 13,608 | 4576 | 6062 |
| Dos | b12,264 | 9237 | 3027 | 4089 |
| Reconnaissance | b10,491 | 7875 | 2616 | 3496 |
| Analysis | b2000 | 1477 | 523 | 677 |
| Shellcode | b1133 | 854 | 279 | 378 |
| Backdoor | b1746 | 1330 | 4116 | 583 |
| worms | b30 | 99 | 31 | 44 |

the KDDCup 99 and NSL-KDD data sets. The UNSW-NB15 contains 42 features. In the dataset there are 42 features, 3 instances are nominal and 39 are numeric features including data types integer, float, and binary . The UNSW-NB15 is divided into two main datasets: UNSW-NB15TRAIN, which is used for training and the UNSW-NB15-TEST which is employed for testing the trained models. In this research we further split the UNSW-NB15-TRAIN into two sets: the UNSW-NB15-TRAIN-1 (75% of the full training data set) for training and the UNSW-NB15-VAL (25% of the full training data set) for validation before testing. The UNSW-NB15 contains samples with nine categories of network attacks: Backdoor, Shellcode, Reconnaissance, Worms, Fuzzers, DOS, Generic, Analysis, Shellcode, and Exploits. Table 2 illustrate the details and the distribution of the values of each attack class within the data subsets.

5.3 Data set CICIDS 2017

The CICIDS 2017 data set '[6] is data developed by the Faculty of Computer Science, the University of New Brunswick in 2017. The data set comprises both benign traffic and details of up-to-date common attacks: such as Brute Force FTP, Brute Force SSH, Dos, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS [43]. This data set shows real generalization of traffic in real networks as well as newer means of attack, we opt the CICIDS 2017 dataset because it consists of 5 days of data collection and the CSV version of CICIDS 2017 contains 2,830,743 rows divided into 8 files, each row having 79 features. Each row of CICIDS 2017 is labeled as Benign or one of fourteen types of attack. In order to create a training and test subset. In the CICIDS 2017 data set, the attack simulation is divided into seven categories including Brute Force Attack, Heart Bleed Attack, Botnet, DoS Attack, DDoS Attack, Web Attack, and Infiltration Attack. CICIDS 2017 has more complex types of attacks as presented in Table 3. The rationale for selecting CICIDS 2017 data set is to have a data set that shows clearly the current real-world network traffic in the experiments.

**Table 3** Attack types in Data set CICIDS 2017

|  | Category | total | training | Test |
|---|---|---|---|---|
| BENIGN | BENIGN | 2,273,097 | 20,000 | 20,000 |
| DOS | DDOS | 128,027 | 2700 | 3300 |
|  | DOS slowloris | 5796 | 1350 | 1650 |
|  | DOS slowhttptest | 5499 | 2171 | 1169 |
|  | DOS hulk | 231,073 | 4500 | 5500 |
|  | DOS goldeneye | 10,293 | 1300 | 700 |
|  | heartbleed | 11 | 75 | 5 |
| portscan | portscan | 158,930 | 3808 | 4192 |
| bot | bot | 1966 | 3936 | 624 |
| brute-force | ftp-patator | 7938 | 900 | 1100 |
|  | ssh-patator | 5796 | 1350 | 1650 |
| web-Attack | web-Attack-bruteforce | 1507 | 910 | 490 |
|  | web-Attack-XSS | 652 | 480 | 160 |
|  | web-Attack-sql-injection | 21 | 16 | 4 |
| infiltration | infiltration | 36 | 24 | 6 |
| total Attack |  | 471,454 | 20000 | 20000 |
| total |  | 2,830,743 | 40000 | 40000 |

## 5.4 Data preprocessing

Raw data often has problems such as noise, bias, sharp changes in dynamic range and sampling, and using them in this way will weaken subsequent designs. Data preprocessing involves all conversions, such as reducing the size of the data that is done on the raw data and converts it to a form that can be used for later processing, such as data classification. It makes them simpler and more effective. Since the nature of the features in the data set is discrete and continuous, so to ensure each observation is in a set of numeric values. KDD includes three symbolic features such as protocol type, service, and flag. We do Data Preprocessing in three steps:

1. Data mapping: symbolic feature values were mapped to a numeric label manually for record in train, test, and validation. The values for these features are mapped to numeric values ranging from 1 to M, where M is the total number of each feature symbolic.
2. Eliminate the duplicate packets: The data set may including duplicate packets that might have the same samples. To avoid this overhead, duplicate packets are eliminated which do not have any effect on model training.
3. Identification of class: data set including a class for each record where the class is either normal connection or a type of attack. Each record from the data set belongs to one class from five major classes: normal, probe, DOS, U2R, R2L. The value for each class is mapped to numeric values. The normal class was mapped to number 5, probe to 4, DOS to 3, U2R to 2, R2L to 1. The method of labeling and numerical range of labels are described in Figure. 7.

| Protocol-Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| Value | | Type | Value | | Type | Value | | Type |
| 1 | | Tcp | 2 | | UDP | 3 | | Icmp |

| Service-Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| Value | Type | Value | Type | Value | Type | Value | Type |
| 71 | harvest | 18 | uucp_path | 36 | echo | 54 | netstat |
| 1 | http | 19 | smtp | 37 | klogin | 55 | pop_3 |
| 2 | ftp | 20 | csnet_ns | 38 | sunrpc | 56 | nnsp |
| 3 | stmp | 21 | uucp | 39 | link | 57 | IRC |
| 4 | private | 22 | netbios_dgm | 40 | login | 58 | pop_2 |
| 5 | auth | 23 | urp_i | 41 | kshell | 59 | printer |
| 6 | Z39_50 | 24 | domain | 42 | sql_net | 60 | tim_i |
| 7 | ftp_data | 25 | bgp | 43 | time | 61 | pm_dump |
| 8 | other | 26 | ldap | 44 | hostnames | 62 | red_i |
| 9 | remote_job | 27 | ecr_i | 45 | exec | 63 | netbios_ssn |
| 10 | name | 28 | gopher | 46 | ntp_u | 64 | rje |
| 11 | eco_i | 29 | vmnet | 47 | discard | 65 | X11 |
| 12 | mtp | 30 | systat | 48 | courier | 66 | urh_i |
| 13 | netbios_ns | 31 | http_443 | 49 | nntp | 67 | http_8001 |
| 14 | telnet | 32 | efs | 50 | ctf | 68 | aol |
| 15 | finger | 33 | whois | 51 | ssh | 69 | http_2784 |
| 16 | supdup | 34 | imap4 | 52 | shell | 70 | tftp_u |
| 17 | domain_u | 35 | iso_tsap | 53 | daytime | | |

| Flag-Type | | | | | | | |
|---|---|---|---|---|---|---|---|
| Value | Type | value | Type | Value | Type | Value | Type |
| 11 | OTH | 3 | REJ | 6 | RSTO | 9 | S3 |
| 1 | SF | 4 | RSTR | 7 | S1 | 10 | S2 |
| 2 | S0 | 5 | SH | 8 | RSTOS0 | | |

**Fig. 7** Data mapping

## 5.5 Experimental setup

This experiment is carried out on a personal computer with Intel Pentium Processor 2.40 GHz Intel Core i7 CPU, 16 GB of RAM, and MATLAB 2016a environment for experiment and python for shape.

In this paper, we selected two data sets, namely, NSL-KDD and KDD99, for experimental. We used five data sets in experimental that taken from training and testing KDD99 in experiments, which are the same in size in all experiments. The number of selected data for each class is shown in Table 4. These values are randomly selected from the experimental and educational data listed in Table 1. Finally, the samples are selected for the experiment using the feature selection algorithm based on the ensemble margin. Some experiments are based on k-fold cross-validation to evaluate test data accuracy. Although we remove the dependency of the samples, that gives us the assurance that the results are not random. Therefore, each training data set is divided into five subsets. In practice, ever times, four portions are put together for the training set, and one of the five-part is used as the test set. Finally, the samples

**Table 4** Data set size used in experiments

|          | Normal | Dos  | Probe | U2R | R2L | Total |
|----------|--------|------|-------|-----|-----|-------|
| Training | 1000   | 4000 | 400   | 52  | 100 | 15285 |
| Testing  | 1000   | 4000 | 400   | 52  | 100 | 21926 |

**Table 5** Confusion Matrix

| first            | Positive(Predicted) | Negative(Predicted |
|------------------|---------------------|--------------------|
| Positive(Actual) | TP                  | FN                 |
| Negative(Actual) | FP                  | TN                 |

are selected for the experiment using the feature selection algorithm based on the ensemble margin. The use of the confusion matrix is one of the ways to evaluate classifiers. This matrix keeps the information in brief. Table 5, shows this matrix.

– True positive (TP): A number of anomaly events that are considered as anomaly events correctly
– False Positive (FP): A number of normal events that are considered as anomaly events by mistake.
– True negative (TN): A number of normal events that are considered as normal events correctly.
– False negative (FN): A number of anomaly events that are considered as normal events by mistake.

Several evaluation criteria can be defined with the help of a confusion matrix, which the most important criterion is Accuracy, Recall,F-measure and Precision.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.6 Analysis of results

We listed the results of the experiments on the five data sets for each base and ensemble in Table 6 .The output of the probabilistic nearest neighbor classifiers and the probabilistic support vector machine is given for integration into the Dempster-Shafer classifier. Overall it can be figured out that the proposed method accuracy in most classes is higher than the accuracy of the other methods.
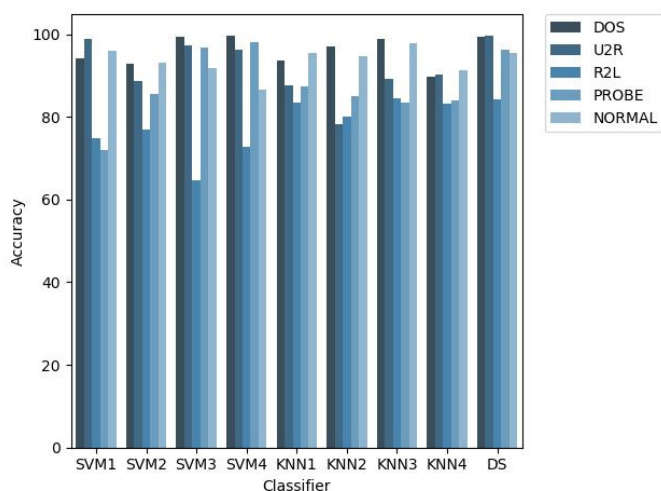
**Fig. 8** Accuracy for data set1

In the method, for U2R attacks, the accuracy in the proposed method is dramatically more than the prior method in which the number of experimental and educational samples is low, the kNN classification has a lower rate of accuracy, but in the proposed method, we achieved higher accuracy using a combination of classifiers. Also, we compared the accuracy rate of the proposed method with other classifiers used in each data set and its results presented for each data set in Table 6. We separately showed the result of the experimental on Figs 8 to Figure 12. These figures illustrated the accuracy of SVM, kNN, and ensemble experts, respectively. In these figures, we show the accuracy of each expert for all data sets.

This Table 7 presents some statistical information about the average results of different experts for data sets used in the proposed method. In general, it can be inferred that the rate of accuracy attack is higher than the other method in other words value obtained for U2R by 99.84% is higher than other attacks.

This experiment aims to illustrate the performance of the proposed methods to increase the accuracy of attacks. Table 9 and Table 11, shows the accuracy of the proposed method with other intrusion detection methods using KDDCUP and NSL-KDD data sets. We used the non-parametric wilcoxon signed-rank test for statistical analysis and comparison of results. The two rows of Table 10 illustrate the results of

**Table 6** Results of accuracy for SVM and kNN and ensemble experts

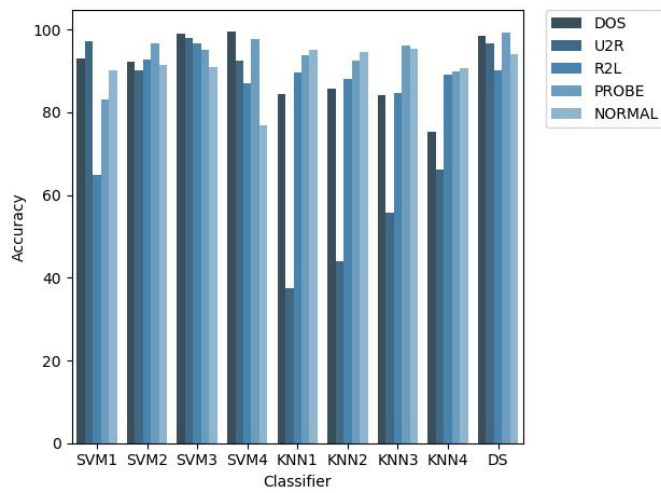| Data set | Classifier | DOS | U2R | R2L | Probe | Normal |
|---|---|---|---|---|---|---|
| | SVM1 | 94.04 | 98.75 | 74.73 | 72.06 | 96.05 |
| | SVM2 | 92.75 | 88.75 | 76.94 | 85.59 | 93.02 |
| | SVM3 | 99.29 | 97.25 | 64.61 | 96.84 | 91.86 |
| | SVM4 | 99.64 | 96.25 | 72.70 | 98.07 | 86.51 |
| Data set 1 | kNN1 | 93.51 | 87.50 | 83.34 | 87.35 | 95.53 |
| | kNN2 | 97.12 | 78.34 | 80.12 | 85.13 | 94.72 |
| | kNN3 | 98.87 | 89.12 | 84.46 | 83.41 | 97.72 |
| | kNN4 | 89.65 | 90.12 | 83.16 | 83.84 | 9128 |
| | DS | **99.60** | **99.73** | **897.95** | **98.16** | **98.53** |
| | SVM1 | 93.14 | 97.12 | 64.78 | 83.12 | 90.16 |
| | SVM2 | 92.28 | 90.13 | 92.90 | 96.78 | 91.41 |
| | SVM3 | 99.06 | 98.02 | 96.70 | 95.23 | 91.03 |
| | SVM4 | 99.66 | 92.43 | 87.16 | 97.62 | 76.91 |
| Data set 2 | kNN1 | 84.34 | 37.50 | 89.63 | 93.78 | 95.05 |
| | kNN2 | 85.82 | 44.12 | 88.03 | 92.54 | 94.56 |
| | kNN3 | 84.16 | 55.78 | 84.79 | 96.16 | 95.45 |
| | kNN4 | 75.34 | 66.18 | 89.10 | 90.02 | 90.80 |
| | DS | **99.74** | **99.92** | **98.49** | **99.28** | **97.56** |
| | SVM1 | 94.83 | 94.12 | 93.16 | 65.71 | 87.13 |
| | SVM2 | 92.16 | 57.61 | 72.83 | 90.33 | 96.88 |
| | SVM3 | 92.83 | 59.78 | 63.15 | 82.12 | 90.17 |
| | SVM4 | 99.48 | 59.78 | 63.15 | 82.12 | 90.17 |
| Data set 3 | kNN1 | 93.40 | 76.47 | 73.47 | 90.66 | 95.90 |
| | kNN2 | 92.65 | 87.33 | 70.18 | 90.27 | 94.55 |
| | kNN3 | 95.53 | 77.42 | 63.08 | 88.48 | 93.81 |
| | kNN4 | 93.35 | 81.34 | 82.16 | 87.26 | 87.74 |
| | DS | **99.68** | **99.77** | **98.22** | **98.31** | **98.46** |
| | SVM1 | 95.83 | 92.16 | 92.83 | 99.48 | 97.12 |
| | SVM2 | 93.18 | 89.12 | 78.16 | 96.88 | 96.18 |
| | SVM3 | 92.37 | 75.16 | 63.15 | 90.17 | 99.31 |
| | SVM4 | 95.42 | 94.38 | 99.31 | 89.45 | 90.47 |
| Data set 4 | kNN1 | 93.40 | 92.12 | 73.21 | 90.66 | 95.90 |
| | kNN2 | 92.65 | 83.33 | 62.17 | 90.27 | 94.55 |
| | kNN3 | 94.53 | 77.42 | 63.08 | 88.48 | 93.81 |
| | kNN4 | 96.42 | 81.34 | 67.22 | 87.26 | 87.78 |
| | DS | **99.75** | **99.85** | **98.78** | **98.16** | **98.18** |
| | SVM1 | 95.93 | 73.45 | 78.80 | 97.73 | 90.16 |
| | SVM2 | 93.42 | 54.16 | 65.21 | 90.29 | 95.32 |
| | SVM3 | 98.64 | 84.12 | 70.73 | 98.11 | 92.98 |
| | SVM4 | 99.63 | 76.45 | 66.12 | 98.49 | 88.13 |
| Data set 5 | kNN1 | 92.98 | 94.16 | 91.30 | 90.58 | 93.81 |
| | kNN2 | 92.40 | 89.50 | 90.51 | 88.23 | 93.62 |
| | kNN3 | 89.54 | 76.12 | 84.58 | 87.81 | 90.85 |
| | kNN4 | 88.32 | 78.36 | 87.60 | 87.64 | 87.31 |
| | DS | **99.68** | **99.96** | **97.77** | **98.97** | **98.46** |

**Fig. 9** Accuracy for data set2



**Fig. 10** Accuracy for data set3

**Table 7** Results of average accuracy for 5 data sets

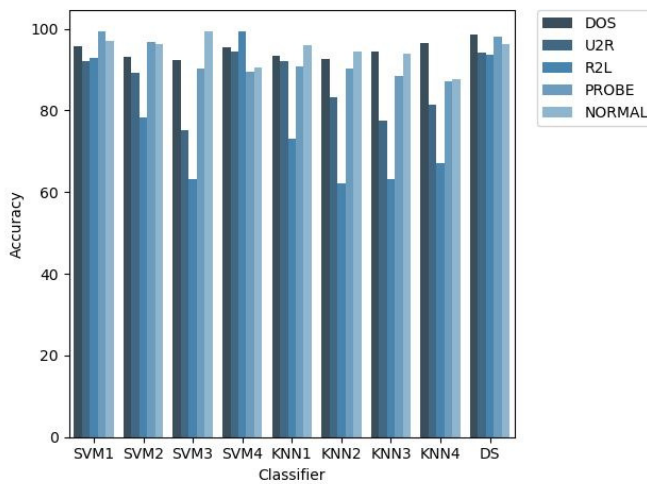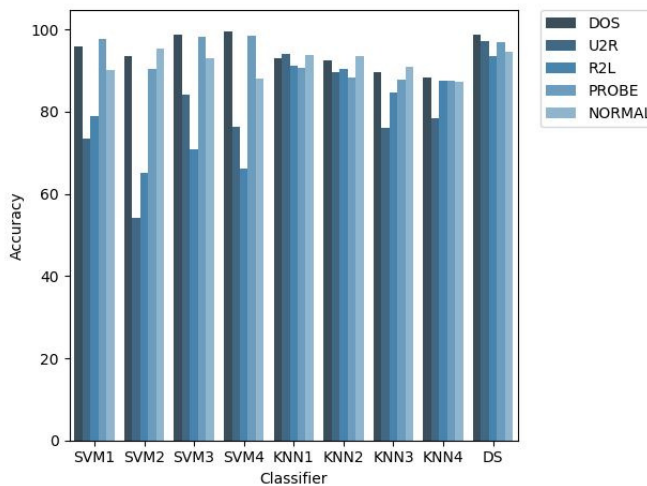| Dataset | DOS | U2R | R2L | Probe | Normal |
|---------|-------|-------|-------|-------|--------|
| Data set1 | 99.60 | 99.73 | 97.95 | 98.16 | 98.53 |
| Data set2 | 96.74 | 99.92 | 98.49 | 99.28 | 97.56 |
| Data set3 | 99.83 | 94.12 | 93.16 | 65.71 | 87.13 |
| Data set4 | 99.75 | 99.85 | 98.78 | 98.16 | 98.18 |
| Data set5 | 99.68 | 99.96 | 99.77 | 98.97 | 98.46 |
| AVG | 99.69 | 98.84 | 98.24 | 98.57 | 98.23 |

**Fig. 11** Accuracy for data set4



**Fig. 12** Accuracy for data set5

the wilcoxon test [8]. Results show the superiority of the classification accuracy of the proposed method in contrast to the other methods.

As can be seen, the accuracy value obtained in the proposed method has especially superiority on Normal, Probe, NORMAL, Dos, and R2L attacks rather than PSO, LUS, and WMA-based methods, and this method do not perform well enough on Normal and R2L attack. Although the proposed method and methods based on WMA significantly increases the accuracy of U2R attacks, CANN, TNN, and DPNN

**Table 8** Performance comparison with feature selection methods.

| Method | Accuracy | precision | F-measure | Recall |
|---|---|---|---|---|
| kNN | 0.97 | 0.89 | 0.98 | 0.93 |
| kNN-Autoencoder | 0.98 | 0.94 | 0.99 | 0.96 |
| kNN-PCA | 0.97 | 0.89 | 0.98 | 0.93 |
| kNN-Variance-threshold | 0.97 | 0.89 | 0.98 | 0.93 |
| kNN-Treebased | 0.97 | 0.89 | 0.98 | 0.93 |

**Table 9** Comparison accuracy of the proposed method with other method

| Method | DOS | U2R | R2L | Probe | Normal | Study |
|---|---|---|---|---|---|---|
| WMA | 98.58 | 99.79 | 82.90 | 96.22 | 95.79 | [1] |
| LUS | 98.85 | 99.80 | 84.72 | 96.88 | 83.50 | [1] |
| PSO | 98.84 | 99.79 | 84.73 | 96.14 | 83.46 | [1] |
| TANN | 90.94 | 60 | 80.53 | 94.89 | 97.01 | [27] |
| DPNN | 99.55 | 55.77 | 88 | 96.50 | 96.50 | [27] |
| CPNN | 99.68 | 3.58 | 88 | 87.61 | 97.04 | [30] |
| The proposed method | **99.69** | **99.84** | **98.35** | **98.57** | **98.23** | our study |

**Table 10** Results of wilcoxon test

| Method | WMA | LUS | PSO | TANN | DPNN | CANN |
|---|---|---|---|---|---|---|
| T value | 0 | 0 | 0 | 0 | 0 | 0 |
| Wilcoxon test | √ | √ | √ | √ | √ | √ |

method [27, 30] have poor performance on R2L and U2R.

As is seen in the diagrams of the accuracy of Figure 13 proposed method has been in the best situation in contrast with other methods, is mentioned above. Somewhat poorer results for the proposed method were obtained for NORMAL by 98.23% that, in contrast with other methods, is higher.

Next, we compare the performance of our method with feature selection methods. To do this, we take all the features of the data set and apply different feature selection techniques, and compare their performance with our method based on autoencoder and the model that uses all the features. Table 8 shows the results of the kNN classifier using all the features, the features generated by autoencoder, and other feature selection techniques. We use Principal Component Analysis (PCA), variance threshold, and tree-based feature selection. We see in the table that our method outperforms all other feature selection techniques, as well as the model that utilizes all the features. This suggests the effectiveness of autoencoder feature extraction technique for this task.

According to Table 11 we can show the superiority of the proposed method approach over listed previous studies in terms of accuracy in IDS using the NSL-KDD data set. In Table 12 we can see the proposed method gain 90.98 accuracy whereas CART, MLP, NB, and CMN achieve 88.67, 89.09, 83.22, 89.95 respectively. Furthermore, for other metrics, the proposed method shows higher results rather than other classifiers for the UNSW-NB15 data set, and NB gives the worst performance in terms of accuracy, precision, and F-measure. As shown in Table 13 our study

**Table 11**  Comparison accuracy of the proposed method with other method using NSL- KDD data set

| Methods | Accuracy | Study |
| --- | --- | --- |
| SVM-ELM | 95.86 | [22] |
| Fusion chi-square and SVM | 98 | [22] |
| HYD | 98.9 | [22] |
| DNN | 81.66 | [13] |
| Ensemble voting | 85.2 | [13] |
| D2H | 99.43 | [4] |
| J48 algorithm | 81.05 | [3] |
| Enhanced J48 algorithm | 90.02 | [3] |
| LSTM-RNN | 93.6 | [9] |
| DBN | 96.91 | [9] |
| DM | 99.80 | our study |

**Table 12**  Performance comparison of the proposed method with other methods on the UNSW-NB15 data set.

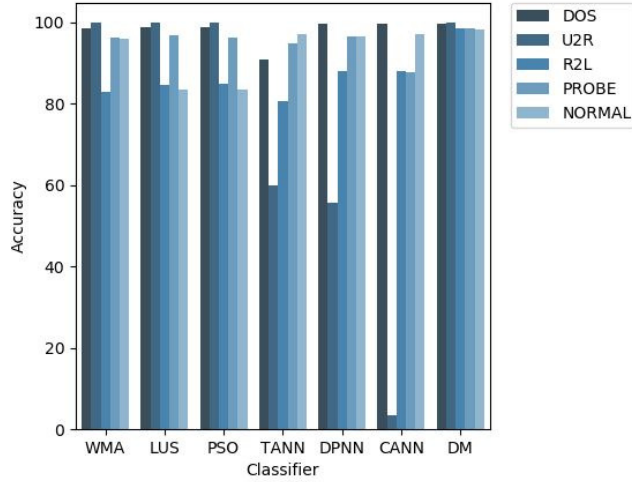| Method | Accuracy | precision | f-measure | Recall | |
| --- | --- | --- | --- | --- | --- |
| CART | 88.67 | 84.32 | 89.04 | 4.32 | [25] |
| MLP | 89.09 | 84.78 | 88.37 | 92.29 | [23] |
| NB | 83.22 | 77.16 | 84.68 | 93.83 | [34] |
| CMN | 89.95 | 82.98 | 90.44 | 99.39 | [43] |
| our study | 90.98 | 87.37 | 93.34 | 99.72 | |

**Table 13**  Performance comparison of the proposed method with other methods on the CICIDS2017 data set

| Method | Accuracy | precision | F-measure | Recall | Study |
| --- | --- | --- | --- | --- | --- |
| RF | 97.89 | 99.92 | 95.63 | 91.70 | [5] |
| kNN | 98.01 | 99.79 | 96.10 | 92.68 | [29] |
| MLP | 97.73 | 99.19 | 96.26 | 93.50 | [23] |
| RKM | 98.04 | 99.86 | 95.99 | 93.29 | [43] |
| our study | 98.97 | 99.90 | 97.08 | 94.43 | |

achieved accuracy of 98.97 on the CICIDS 2017 data set and experts of RF, kNN, MLP and RKM give accuracy of 97.89, 98.01, 97.73, and 98.04 respectively. In this data set, MLP achieves the worst result and our study achieve a better result for all metric rather than other methods. As illustrate in Table 14, the proposed ensemble models show the higher result in terms of accuracy, precision, recall, and F-measure on NLS-KDD data set, with better performance in comparison to other classifiers. We also showed the result of the experimental on Figure 16,Figure 15 and Figure 14. This due to in ensemble method the correct decisions are strengthened, and incorrect decisions are canceled or weakened, and these methods have better performance than single methods in detection attacks.

**Table 14** Performance comparison of the proposed method with other methods on the NSL-KDD data set

| Method | Accuracy | precision | F-measure | Recall | Study |
|--------|----------|-----------|-----------|--------|-------|
| CRT | 98.91 | 98.89 | 98.65 | 98.70 | [25] |
| kNN | 99.63 | 99.69 | 99.63 | 99.59 | [29] |
| MLP | 99.69 | 99.73 | 99.68 | 99.64 | [23] |
| CKM | 99.73 | 99.77 | 99.77 | 99.79 | [43] |
| our study | 99.80 | 99.83 | 99.83 | 99.84 | |



**Fig. 13** Accuracy of the proposed method, compared with other methods.

## 6 Conclusions

In this paper, we proposed a novel method to improve the performance and getting a high accuracy in detecting the attacks network via the ensemble method. As in the ensemble method, the correct decisions are strengthened, incorrect decisions are weakened.

In this work, first, we used from SVM and kNN experts, heuristic function in kNN and sigmoid function used for converting the format of its output to probable output. Finally, Dempster Shafer's used for Combining base experts into an ensemble expert. Deep learning for feature extraction and ensemble margin used for the selected better samples from the data set. Although, in the detection of R2L and Probe attacks, it has been able to achieve the highest accuracy of the common methods in this field.

We performed different experiments on our proposed method and observed the performance of all the classifiers. The results of the experiments on UNSW-NB15, CICIDS 2017, and NSL-KDD data sets showed the superiority of the proposed method
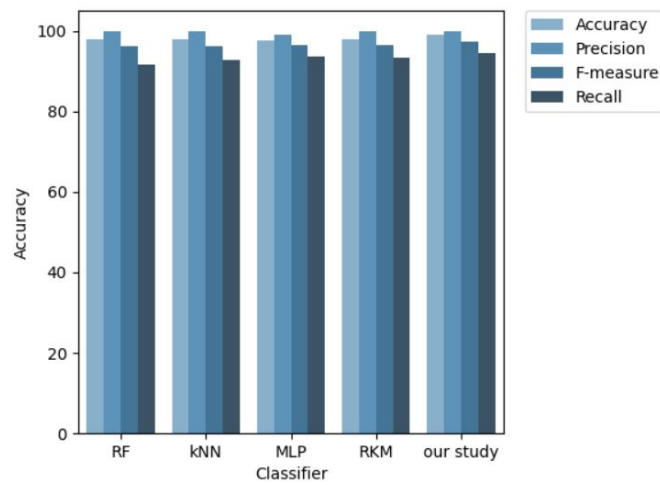
**Fig. 14** Performance comparison of the proposed method with other methods on the CICIDS 2017 data set
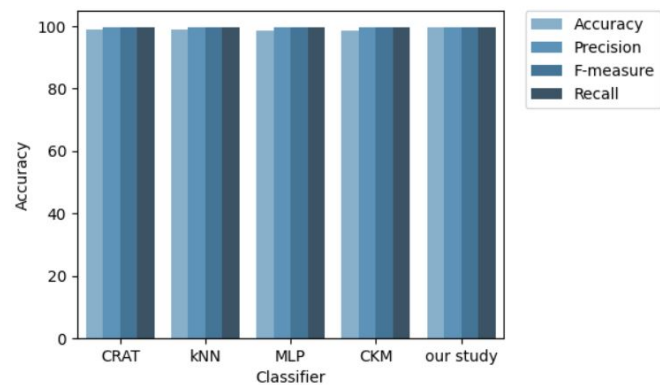


**Fig. 15** Performance comparison of the proposed method with other methods on the NSL-KDD data set

in comparison with the other methods in terms of accuracy, precision, F-measure, and recall. In future works, we will study how to compute the probabilities of the output of SVM and kNN using other methods to reduce time and improve performance. We also tend to use other classifiers such as the Bayesian network, whose output is probable, as one of the basic classifiers which will perform better on accuracy.
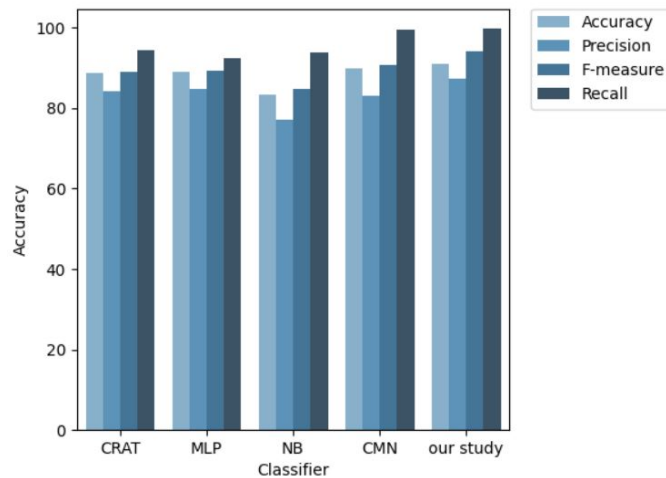
**Fig. 16** Performance comparison of the proposed method with other methods using the UNSW-NB15 data set

Compliance with Ethical Standards

Conflict of Interest: All Authors declare that they have no conflict of interest.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.
Informed consent was obtained from all individual participants included in the study.

# References

[1]   Aburomman AA, Reaz MB. "A novel SVM-kNN-PSO ensemble method for intrusion detection system". In: *Applied Soft Computing* 38 (2016), pp. 360–372.

[2]   Ahmed M, Mahmood AN, Hu J. "A survey of network anomaly detection techniques". In: *Journal of Network and Computer Applications* 60 (2016), pages19–31.

[3]   Aljawarneh S, Yassein MB, Aljundi M. "An enhanced J48 classification algorithm for the anomaly intrusion detection systems". In: *Cluster Computing* 22.5 (2019), Pages 10549–10565. DOI: `10.1007/s10586-017-1109-8`.

[4]   Aloqaily M, Otoum S, Al Ridhawi I, Jararweh Y. "An Intrusion Detection System for Connected Vehicles in Smart Cities". In: *Ad Hoc Networks* (2019). DOI: `https://doi.org/10.1016/j.adhoc.2019.02.001`.

[5]   Breiman L. "Random forests". In: *Machine learning* (2001), Pages 5–32.

[6]   *CICIDS2017dataset2018*. URL: `https://www.unb.ca/cic/datasets/ids-2017.html/.AccessedJanuary2,2019.`.

[7]   Cover T, Hart P. "Nearest neighbor pattern classification". In: *EEE transactions on information theory* 13 (1967), pages 21–27. DOI: `10.1109/TIT.1967.1053964`.

[8]   Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *Journal of Machine learning research* (2006), Pages 1–30.

[9]   Elmasry W, Akbulut A, Zaim AH. "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic". In: *Computer Networks* 168 (2020). DOI: `https://doi.org/10.1016/j.comnet.2019.107042`.

[10]  Al-Enezi KA, Al-Shaikhli IF, Al-Kandari AR, Al-Tayyar LZ. "A Survey of Intrusion Detection System Using Case Study Kuwait Governments Entities". In: *International Conference on Advanced Computer Science Applications and Technologies* (2014). DOI: `10.1109/ACSAT.2014.14`.

[11]  Folino G, Pisani FS, Sabatino P. A. "A Distributed Intrusion Detection Framework Based on Evolved Specialized Ensembles of Classifiers". In: *in Applications of Evolutionary Computations* (2016), pp. 315–331.

[12]  Shafer G. *A Mathematical Theory of Evidence*. London: Princeton Universityy, 1976.

[13]  Gao X, Shan C, Hu C, Niu Z, Liu Z. "An Adaptive Ensemble Machine Learning Model for Intrusion Detection". In: *IEEE Access* 7 (2019), Pages 82512–82521. DOI: `10.1109/ACCESS.2019.2923640`.

[14]  Gautam SK, Om H. "Anomaly detection system using entropy based technique". In: *1st International Conference on Next Generation Computing Technologies (NGCT)* (2016). DOI: `10.1109/NGCT.2015.7375219`.

[15]  Günther J, Pilarski PM, Helfrich G, Shen H, Diepold K. "Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning". In: *Nature* 34 (2016), pages 1–11. DOI: `https://doi.org/10.1016/j.mechatronics.2015.09.004`.

[16]  Hamidzadeh J, Moslemnejad S. "Identification of uncertainty and decision boundary for SVM classification training using belief function". In: *Applied Intelligence* 49 (2019).

[17]  Javad Hamidzadeh, Reza Monsefi and Hadi Sadoghi Yazdi. "Large symmetric margin instance selection algorithm". In: *International Journal of Machine Learning and Cybernetics* 7.1 (2016), pages25–45.

[18]  Javid M, Hamidzadeh J. "An active multi-class classifcation using privileged information and belief function". In: *International Journal of Machine Learning and Cybernetics* (2019), pages1–14. DOI: `https://doi.org/10.1007/s13042-019-00991-w`.

[19]  Anderson JP. "computer Security Threat Monitoring and Surveillance". In: *International Journal of Computer Science and Mobile Computing* (1908).

[20]  Kaushik SS, Deshmukh PR. "Detection of Attacks in an Intrusion Detection System". In: *International Journal of Computer Science and Information Technologies* 2 (2011), pages982–986.

[21]  Keramati A, Jafari-Marandi R, Aliannejadi M, Ahmadian I, Mozaffari M, Abbasi U. "Improved churn prediction in telecommunication industry using data mining techniques". In: *Applied Soft Computing* 24 (2014), pages994–1012.

[22]  Khonde SR, Ulagamuthalvi V. "Ensemble-based semi-supervised learning approach for a distributed intrusion detection system". In: *Journal of Cyber Security Technology* 3.3 (2019), Pages 163–188. DOI: `https://doi.org/10.1080/23742917.2019.1623475`.

[23]  Kubat M. "Neural networks: a comprehensive foundation by Simon Haykin". In: *The Knowledge Engineering Review* 13.4 (1999), Pages 409–412. DOI: `https://doi.org/10.1017/S0269888998214044`.

[24]  Kumari VV, Varma PR. "A semi-supervised Intrusion Detection System using active learning SVM and fuzzy c-means clustering". In: *International conference on I- SMAC* (2017). DOI: `10.1109/I-SMAC.2017.8058397`.

[25]  Breiman L. *Classification and Regression Trees*. NewYork: Routledge, 2017.

[26]  LeCun Y, Bengio Y, Hinton G. "Deep learning". In: *Nature* (2015), pages 436–444.

[27]  Li L, Zhang H, Peng H, Yang Y. "Nearest neighbors based density peaks approach to intrusion detection". In: *Expert Systems with Applications* 110 (2018), pp. 33–40. DOI: `https://doi.org/10.1016/j.chaos.2018.03.010`.

[28]  Li W, Meng W, Kwok LF, Horace HS. "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model". In: *Journal of Network and Computer Applications* 77 (2017), pages135–145.

[29]  Liao Y, Vemuri VR. "Use of k-nearest neighbor classifier for intrusion detection". In: *Computers AND security* 21.5 (2002), Pages 439–448.

[30]  Lin WC, Ke SW, Tsai CF. CANN. "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors". In: *Knowledge-based systems* 78 (2015), Pages 13–21. DOI: `https://doi.org/10.1016/j.knosys.2015.01.009`.

[31]  Ludwig SA. "Applying a Neural Network Ensemble to Intrusion Detection". In: *Journal of Artificial Intelligence and Soft Computing Research* 9.3 (2019). DOI: `https://doi.org/10.2478/jaiscr-2019-0002Openaccess`.

[32]  Moghaddam VH, Hamidzadeh J. "New Hermite orthogonal polynomial kernel and combined kernels in Support Vector Machine classifier". In: *Pattern Recognition* 60 (2016), pages 921–935.

[33]  Moustafa N, Hu J, Slay J. "A Holistic Review of Network Anomaly Detection Systems: A Comprehensive Survey". In: *Journal of Network and Computer Applications* 128 (2019), pages33–55.

[34]  Murphy KP. "Naive Bayes Classifiers". In: *Security and Privacy Issues in Sensor Networks and IoT* 18 (2006).

[35]  Naphade MR, Raut MP, Dande AA. "A Review of Intrusion Detection System Basic Concepts". In: *International Journal of Computer Science and Mobile Computing* 5.3 (2016), pp. 482–485.

[36] Park TJ, Chang JH. "Dempster-Shafer D2 theory for enhanced statistical model-based voice activity detection". In: *Computer Speech & Language* 47.3 (2018), pages47–58. DOI: `https://doi.org/10.1016/j.csl.2017.07.0012`.

[37] Platt J. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* (1999).

[38] Saidi M, Bechar ME, Settouti N, Chikh MA. "Instances selection algorithm by ensemble margin". In: *Journal of Experimental & Theoretical Artificial Intelligence* 30.3 (2018), pages457–478. DOI: `https://doi.org/10.1080/0952813X.2017.1409283`.

[39] El-Sappagh S, Mohammed AS, AlSheshtawy TA. "Classification Procedures for Intrusion Detection based on KDD CUP 99 Data Set". In: *International Journal of Network Security* 11.3 (2019), pages41525–41550. DOI: `10.5121/ijnsa.2019.11302`.

[40] Schapire RE, Freund Y, Bartlett P, Lee WS. "Boosting the margin: a new explanation for the effectiveness of voting methods". In: *The annals of statistics* 26.5 (1998), pages1651–1686.

[41] Schmidhuber J. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pages85–117. DOI: `https://doi.org/10.1016/j.neunet.2014.09.003`.

[42] Singh R, Kumar H, Singla RK. "An intrusion detection system using network traffic profiling and online sequential extreme learning machine". In: *Expert Systems with Applications* 42.22 (2015), pp. 8609–8624.

[43] Swami R, Dave M, Ranga V. "Voting-based intrusion detection framework for securing software-defined networks". In: *Concurrency and Computation: Practice and Experience* (2020).

[44] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. "A detailed analysis of the KDD CUP 99 data set". In: *IEEE Symposium on Computational Intelligence for Security and Defense Applications* (2009), pages53–58. DOI: `10.1109/CISDA.2009.53565284`.

[45] *UNSW-NB15dataset2017*. URL: `.https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.AccessedOctober19, 2018.`.

[46] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. "Deep Learning Approach for Intelligent Intrusion Detection System". In: *IEEE Access* 7 (2019), pages41525–41550. DOI: `10.1109/ACCESS.2019.2895334`.

[47] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". In: *Journal of machine learning research* (2010), pages 3371–3408.

[48] Zabihi M, Jahan MV, Hamidzadeh J. "A density based clustering approach for web robot detection". In: *4th International Conference on Computer and Knowledge Engineering (ICCKE)* (2014). DOI: `10.1109/ICCKE.2014.6993362`.

[49] Zaman K, Rangavajhala S, McDonald MP, Mahadevan S. A. "A probabilistic approach for representation of interval uncertainty". In: *Reliability Engineer-*

*ing & System Safety* 96 (2011), Pages 117–130. DOI: `https://doi.org/10.1016/j.ress.2010.07.012`.

[50] Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC. "A survey of intrusion detection in Internet of Things". In: *Journal of Network and Computer Applications* 84 (2017), pages25–37. DOI: `https://doi.org/10.1016/j.jnca.2017.02.009`.

[51] Zhang H, Huang L, Wu CQ, Li Z. "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset". In: *Computer Networks* (2020).

[52] Zhang Y, Liu B, Cai J, Zhang S. "Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution". In: *Neural Computing and Applications* 28 (2017), pages259–267.