A novel method combining fuzzy SVM and sampling for imbalanced classification

Tao Ma and Ying Hou

Department of Information Science and Engineering,
Lanzhou University,
LanZhou 730000, China
Email: mat13@lzu.edu.cn
Email: houy15@lzu.edu.cn

Jian-Jun Cheng

Department of Information Science and Engineering,
Lanzhou University,
LanZhou 730000, China
and
Gansu Resources and Environmental Science Data Engineering
Technology Research Center,
Lanzhou 730000, China
Email: chengjianjun@lzu.edu.cn

Xiao-Yun Chen*

Department of Information Science and Engineering,
Lanzhou University,
LanZhou 730000, China
Email: chenxy@lzu.edu.cn
*Corresponding author

Abstract: The class imbalance problem has been reported to reduce performance of many existing learning algorithms in intrusion detection. However, the detection rates for minority classes still need to be improved. Thus, the novel hybrid method FSVMs is proposed to solve the problem in the paper, which integrates the prevailing sampling method SMOTE with fuzzy semi-supervised SVM learning approach to class imbalanced intrusion detection data. The basic KDD Cup 1999 dataset, NSLKDD dataset and imbalanced dataset from UCI are used to evaluate the performance of proposed model. Experiment results show that the proposed method outperforms other state-of-the-art classifiers including support vector machine (SVM), back propagation neural network (BPNN), Bayes, k-nearest neighbour (KNN), decision tree (DT), random forest (RF) and four sampling methods in the aspects of detection rate and false alarm rate, and has better robustness for imbalanced classification.

Keywords: intrusion detection; fuzziness; support vector machine; SVM; SMOTE; semi-supervised learning; SSL; imbalance classification; applied systemic studies.

Reference to this paper should be made as follows: Ma, T., Hou, Y., Cheng, J-J. and Chen, X-Y. (2018) 'A novel method combining fuzzy SVM and sampling for imbalanced classification', *Int. J. Applied Systemic Studies*, Vol. 8, No. 1, pp.1–31.

Biographical notes: Tao Ma received his PhD in Computer Science from the School of Information Science and Engineering, Lanzhou University in 2017. He is an Associate Professor at the School of Mathematical and Computer Science of Ningxia Teachers University. His research interests include data mining and algorithm optimisation.

Ying Hou is currently a graduate student in the School of Information Science and Engineering, Lanzhou University. Her research interest is classification and prediction in the field of data mining.

Jian-Jun Cheng earned his doctorate degree in 2015 and works as a Lecturer at the LanZhou University. In his doctoral dissertation, he studied the community network detection and clustering analysis.

Xiao-Yun Chen is a Professor at the Institute of Information in Lanzhou University and is a senior member of China Computer Society. Her research direction includes data mining, data warehouse, web mining, and meteotological information processing. She has presided over more than ten research projects.

This paper is a revised and expanded version of a paper entitled 'A hybrid methodologies for intrusion detection based deep neural network with support vector machines and clustering techniques' presented at The 5th International Conference on Frontier Computing (FC 2016), Tokyo, 13–15 July 2016.

1 Introduction

With the development of network technology, people are more relying on the network to work and study. The research and development of intrusion detection technology become more popular in the field of machine learning research (Ojala, 2013). Intrusion detection is a kind of security mechanism; it can detect abnormal access for ensuring the safety of computer and network communication.

Machine learning algorithms were broadly used in intrusion detection. Some basic learning methods become the first choice of researchers, including DT (Eesa et al., 2015), RF (Hasan et al., 2014), Bayesian network (Devarakonda et al., 2012; Koc et al., 2012), Markovian (Huang et al., 2013), expectation maximisation (Hamed and Hamid, 2014), support vector machine (Kim et al., 2014), KNN (Canbay and Sagiroglu, 2015), extreme learning machine (Fossaceca et al., 2015) and so on. Although they have better performances in binary classification or balanced classification problem, they tend not to perform well on imbalanced classification, as showed in Table 1.

For multi-classification problem in intrusion detection, though previous reports showed that SVM had better classification ability than other classifiers, but it has some weak points. Therefore, hybrid classifiers model based on SVM were put forward to improve performance and robust of single method, Cijo et al. (2013) proposed local deep kernel learning (LDKL) to speed up nonlinear SVM prediction and accuracy, but it only

was performed for two-class classification. The methods, combining KNN (Aburomman and Ibne Reaz, 2016), DT (Ji et al., 2016), MCLP (Hosseini Bamakan et al., 2016), had been used to improve the accuracy and false alarm rate of classification. Some researchers (Karami and Guerrero-Zapata, 2015; Ahmed et al., 2016; Bhuyan et al., 2016; Zhang et al., 2015) made contributions to the anomaly detection method's survey in IDS. These methods have many advantages in dealing with multi-classification problem. However, there are some drawbacks in tackling imbalanced problem and new attack types. Therefore, some researchers introduced clustering algorithms to improve performance.

Table 1 Comparison of algorithms learning strategies and difficulties

Method	*Strategy*	*Difficulties*
Decision tree	Splitting the training data and assigning label to leaf recursively, and pruning a leaf or a branch	Many splits for small class, branches or leaves for small class may be pruned
BP neural networks	Adjusting the weights between neurons by the gradient descent	Prevalent class dominate the gradient descent direction
Bayesian	Exploring dependency patterns among attributes	Dependency patterns in the small class are hard to be encoded
Support vector machines	Finding an optimal separating hyperplane to maximise the margin and minimise the training error	Support vectors of the small class are rare, decision boundary are skewed toward the small class
K-nearest neighbour	Deciding the label of sample by the maximum class within the k nearest neighbours	The k nearest neighbours of the prevalent class samples bear higher probabilities

Clustering technologies were used to reduce irrelevant dimensional and improve the detection rate of attack classes in the past studies. Feng et al. (2014) combined the ACN clustering into SVM for improved classification rate. Ravale (2015) used a classification module to realise anomaly intrusion by combining K-means clustering algorithm and SVM in the KDD CUP'99 dataset. In the paper (Lin et al., 2015), the author introduced the clustering and KNN method to reduce irrelevant dimensional for improving attack's detection rate. Though unlabelled data can be classed effectively by using combined clustering methods, the imbalanced problem was not resolved well.

Depending on the number distribution of different class, the classification has two types: Balanced and Imbalanced. Imbalanced data is defined as the dataset that some class is larger than any class in number, the larger class is described as the majority or negative class and the less class is described as minority or positive class. As rare instances occur infrequently, classification rules tend to be ignored; test samples belonging to the small classes often is misclassified.

So, research on the imbalanced problem is critical in data mining and machine learning. However, some basic algorithms have learning difficulties in imbalanced dataset. It can be briefly shown in Table 1.

The basic KDD Cup'99 and NSLKDD dataset are extensively used in the field of intrusion detection; however, the dataset is imbalanced for five classes: the number of R2L and U2R are more litter than other attacks. Moreover, the common classification algorithms have difficulty in dealing with the situation, rare class may be marked noise and dropped, their distinguish rate lag far behind general classes. Models trained by the

imbalanced data often have low prediction accuracy in the minority or overall, but, correct identification of each kind of attack is equally important for intrusion detection system (IDS). Therefore, the realisation of multiclass of imbalanced classification, the selection of feasible solutions and the proper evaluating measures become experts' goals in constructing model for intrusion detection.

For dealing with the imbalanced problem, some researchers have taken measures that can be divided into three types:

1 ensemble methods (Sun et al., 2015)

2 algorithmic approaches (Zhang et al., 2016): modification learning process of algorithm

3 sampling approaches (García et al., 2016): it is considered to rebalance the data by sampling in data level.

According to the recent study (García et al., 2016) of imbalanced data classification, sampling is more effective than other approaches. Therefore, the sampling methods are made a study in this paper.

Sampling method aims to sample training sets with different ratios for the majority class and the minority class to control the size of them. There are three ways to extract sample from the original datasets: randomly under-sampling (RUS) is referred to randomly remove the number of majority instances for balancing dataset, but this method can potentially discard useful data (Tahir et al., 2012). Randomly over-sampling (ROS) is a method that randomly duplicating the existing data or creating synthetic data for minority classes, it can be shown that over-sampling is generally more outperformed than under-sampling in previous experimental results (García et al., 2012). The researchers are interested in combination of under-sampling and over-sampling. Synthetic minority oversampling technique (SMOTE) is an intelligent hybrid sampling approach, it creates the synthetic minority samples by finding the k nearest neighbours of each instance, then it can reduce the data samples by judging whether the sample and its five neighbours have the same label. Sain et al. have developed sampling ensemble algorithms based on SVM classifier (Sain and Purnami, 2015; Jian et al., 2016).

To better tackle imbalanced dataset, the semi-supervised techniques are considered to overcome the disadvantages of the adoption of supervised and unsupervised learning solely in IDS. Because supervised learning approaches only use labelled samples to train a classifier, obtaining labelled samples is difficult and requires abundant efforts of domain experts, but unlabelled data can easily be obtained in many real world problems. Some unsupervised learning methods have been adopted in the papers to resolve the intrusion detection problem (De La Hoz et al., 2014, 2015). Compared to supervise learning approaches, semi-supervised learning (SSL) is outperformed by considering a larger number of unlabelled samples together with the labelled samples to build a classifier in the aspect of addressing the kind issue (Zhu and Goldberg, 2009). Nekooeimehr and Lai-Yuen (2016) presented adaptive semi-supervised weighted oversampling (A-SUWO) model for imbalanced binary datasets classification. The fuzziness based on SSL algorithm proposed by Ashfaq et al. (2016) is used to improve the classifier's performance for IDSs.

From the above, the sampling and semi-supervised methods have drawbacks in class imbalanced data: sampling method adds the probability of over-fitting, semi-supervised method hasn't efficient performance for higher imbalance rate. In order to solve the

above problems, in this paper, the FSVMs is proposed which combining fuzzy SVM and SMOTE sampling algorithm to deal with imbalanced multi-classification in intrusion detection. The main contributions of the paper:

1 A new SSL algorithm is designed for improving the classifier performance on IDS datasets by investigating a divide-and-conquer strategy, unlabelled samples with their predicted labels are categorised according to the magnitude of fuzziness.

2 The support vector machine with weights (SVMw) is served as based classifier because it thinks about the weight of each sample, it has a better learning performance and it is computationally efficient.

3 For tackling the multiclass imbalance problem, the novel sample method mix-ratio SMOTE was used in data pre-processing stage, it not only can realise the efficient sampling, but also can avoid over-fitting phenomenon.

The remainder of the paper is organised as follow. Some related works are discussed in Section 2. The proposed method is depicted in Section 3, including the framework, principles and procedures. In Section 4, the main work includes the data preparation and evaluation criteria. Experiment results and some discussions in Section 5. Conclusions and future research directions are pointed out in Section 6.

2 Related work

2.1 Mix-SMOTE sampling

2.1.1 Synthetic minority oversampling technique

SMOTE is one of oversampling methods. It applies sampling method to increase the number of positive class by creating randomly data replication or synthetic data, so that the amount of positive data is almost equal to the negative, SMOTE algorithm was first introduced by Chawla et al. (2002). SMOTE algorithm defines k nearest neighbours for each positive class rapidly; then constructs the data synthetic reproduction between positive class and the randomly chosen k nearest neighbours. Generally it can be expressed in formula as follow equation (1).

$$x_{syn} = x_i + (x_{knn} - x_i) \times \delta \tag{1}$$

where δ is a random number between 0 and 1.

These sampling approaches mentioned above have shown their efficiency in dealing with the imbalanced dataset. However, they are proposed for binary classes only, there are one majority class and one minority class.

In multi-classification problem, especially when the number of classes is large, applying similar sampling procedures is questionable, because it is difficult to distinguish majority from the minority class. Assume there are n classes, the largest class and the smallest class can be regarded as the majority class and the minority class respectively, then the rest $n - 2$ classes can be either majority or minority. Therefore, there are 2^{n-2} possible cases in dividing the classes into two groups. Simply applying the over-sampling or under-sampling techniques mentioned above is computational expensive and

unmanageable. There is a need of intelligent sampling which can provide an effective sampling procedure in the multiclass imbalance problem.

2.1.2 Mix-ratio sampling

It is first proposed in the paper by Bae et al. (2010). The sampling algorithm is briefly described as follows: a multiclass classifier is implemented on the original training set (T) and its classification performance for each class is kept as the baseline (C_T). Next, different models with sampling ratios are implemented. The training set (T) is updated by extracting the samples from each class. Performance matrix $C(i, j)$ represented the classification result for class j using sampling ratio i. If the best performance ($\max(C(i, j))$) for a class is better than the baseline ($C_T(j)$), then the minority classes and their over-sampling ratios are determined.

Based on this thought of mix-ratio sampling method, we design a mix-SMOTE sampling for KDD dataset, it has three minority classes (probe, U2R and R2L); the sampling ratio of each class is determined according to the percentage of each minority class and the second majority class. This way is selected because the performance is better than other ratios. Through the experiment, it can be verified that the ratio sampling can effectively solve multiclass imbalanced problem in KDD dataset, even though the number of one class is larger than others.

2.2 Fuzziness

The term was first mentioned by Zadeh (1968), it refers to the unclear boundary, Zadeh also generalised the probability measure of an event to a fuzzy event and using entropy in information theory. The author De Luca considered fuzziness to be a type of uncertainty and also define a measure of fuzziness with non-probabilistic entropy similar to Shannon's information entropy in paper (De Luca and Termini, 1972). The fuzziness should hold the properties: the fuzziness degree should attain its maximum when the membership degree of every element is equal and its minimum when every element either belongs to the fuzzy set or absolutely not. In this literature (Sánchez and Trillas, 2012), the fuzziness of a fuzzy set can be calculated by a function $F \rightarrow [0, 1]^x$, it meet the axioms shown as follow:

1 $F(\mu) = 0$ if and only if μ is a crisp set.

2 $F(\mu)$ gets its maximum value, if and only if $\mu(x) = 05 \forall x \in X$.

3 If $\mu \leq_s \sigma$, then $F(\mu) \geq F(\sigma)$.

4 $F(\mu) = F(\mu')$ where $\mu'(x) = 1 - \mu(x)$ for $\forall x \in X$.

5 $F(\mu \cup \sigma) + F(\mu \cap \sigma) = F(\mu) + F(\sigma)$.

Axiom (1–3) were already put forward by De Luca and Termini (1972). For we can measure the fuzziness, it is needed for us to know when a fuzzy set is less than another. The order is defined as follows:

$$\begin{aligned} \mu \leq_S \sigma <=> & \min(0.5, \mu(x)) \geq \min(0.5, \sigma(x)) \\ & \& \max(0.5, \mu(x)) \leq \max(0.5, \sigma(x)) \end{aligned} \tag{2}$$

Definition 1: assuming U = $\{\mu_1, \mu_2, \ldots, \mu_n\}$ is a fuzzy set. According to De Luca and Termini(1972), the fuzziness of U can be formulated as:

$$F(U) = -\frac{1}{n}\sum_{i=1}^{n}\left(\mu_i \log \mu_i + (1-\mu_i)\log(1-\mu_i)\right) \tag{3}$$

Some research constructed equations similar to the above: when $n = 2$ and U is normalised according to $\mu_1 + \mu_2 = 1$, then we have

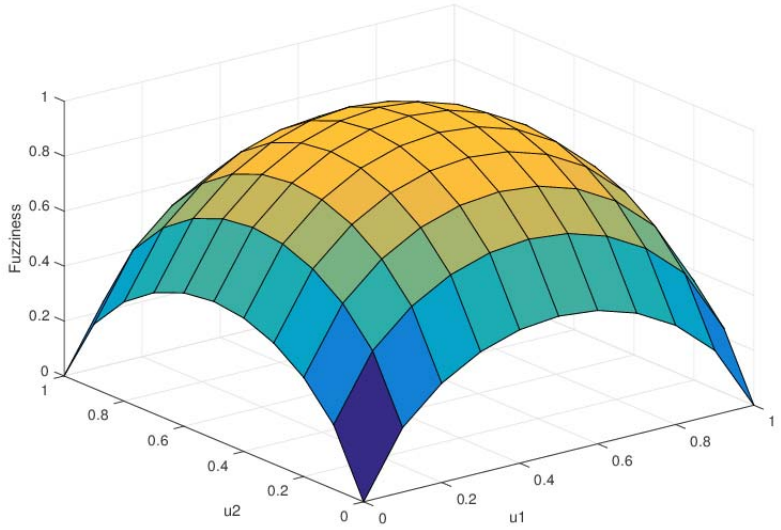
$$F_1(U) = 1 - \mu_1^2 - (1-\mu_1)^2 \tag{4}$$

$$F_2(U) = \begin{cases} \dfrac{\mu_1}{1-\mu_1} & 0 \le \mu_1 \le 0.5 \\ \dfrac{1-\mu_1}{\mu_1} & 0.5 \le \mu_1 \le 1 \end{cases} \tag{5}$$

It has been verified that the fuzziness of fuzzy set gets its maximum when the membership degree is $\mu_i = 0.5$ for each $i = 1, 2, \ldots, n$ and minimum when each $\mu_i = 0$ or $\mu_i = 1$, $i(1 \le i \le n)$, it represents the element belongs to the fuzzy set or not.

equation (3) can be depicted as Figure 1, where the fuzziness has its maximum at the point (0.5, 0.5) and minimum at (0, 0),(0, 1),(1, 0),(1, 1) for a binary class problem.

Figure 1 Fuzziness for binary class (see online version for colours)



When connect the fuzziness of a fuzzy vector with a classifier output. It is found that many classifiers have got a membership degree of testing instance belonging to each class (C1, C2, C3, C4, C5) in which each component corresponds to an output manner of fuzzy vector. The kind of classifiers includes neural network, support vector machine, fuzzy decision trees, KNN, extreme learning machine, etc.

For a given training set $\{X_i\}_{i=1}^{N}$, the membership degree of every instance belongs to a particular class C is got. The membership vector matrix, $U = (U_{ij})_{(C\times N)}$, denotes the fuzzy partition of all samples. The components of the matrix must meet the limited condition shown as follow:

$$\sum_{i=1}^{C} \mu_{ij} = 1,\ 0 < \sum_{j=1}^{N} \mu_{ij} < N,\ \mu_{ij} \in [0,1] \tag{6}$$

where $\mu_{ij} = \mu_i(x_j)$ represents the membership degree of the j^{th} sample x_j belongs to the i^{th} class. For classifier, the membership matrix U can be obtained at the end of training phase. In testing process, the classifier can give an output for each j^{th} sample in the form of fuzzy vector. On the basic of equation (3), the fuzziness of every output vector can be formulated as:

$$F(\mu_j) = -\frac{1}{C}\sum_{i=1}^{C}\left(\mu_{ij}\log\mu_{ij} + (1-\mu_{ij})\log(1-\mu_{ij})\right) \tag{7}$$

2.3 Divide-and-conquer strategy

In recently paper, the authors have proposed an algorithm based on a divide-and-conquer strategy (Wang et al., 2015). This method can be understood as a SSL process, in which the training data includes many unknown label samples having low fuzziness. The core parts of the algorithm proposed by Wang et al. (2015) are shown in Table 2.

Table 2 Fuzziness based divide-and-conquer strategy

Given training set *Tr*, testing set *Te*, and classifier *C*	
1	Get the fuzzy membership vector output by using the classifier to *Te*.
2	Compute the fuzziness in every output vector, and obtain the accuracy Tr_{acc} and Te_{acc}.
3	All testing samples *Te* were divided into three groups (low, mid, high) according to the fuzziness
4	The group with highest accuracy was incorporated into the original training set *Tr*, then get new dataset *nTr* for training.
5	Perform retraining with the new training set *nTr*, and get the training accuracy nTr_{acc} and testing accuracy Te_{acc}.
6	Compare the accuracies obtained above.

2.4 Support vector machines

2.4.1 SVM for binary classification

The fundamental goal of SVM is to construct the optimal separating hyperplane, it is decided by support vectors. We define it as the decision function which gives the maximum separation margin between two classes.

In a binary classification problem, considering a training set $x_i \in R^n$ and its label set $y_i \in \{+1, -1\}$, in which $y_i = +1$ represents positive class and $y_i = -1$ represents negative class for all training data $i = 1, 2, 3, \ldots, m$, where m is the number of the data and n is the

dimension of the data. The hyperplane $g(x)$, that splits the given data, is defined as equation (8):

$$g(x) = \sum_{i}^{m} w_i x_i + b = W^T X + b \tag{8}$$

where W is the n dimensional weight vector, b is a bias term. The W and b describe the shape and position of hyperplane respectively. The distance between a hyperplane and the data point nearest to the hyperplane is the margin calculated by $2 / |W|$. Training SVM model for finding the optimal hyperplane which has the maximum margin, meeting the following constraint:

$$y_i \left(W^T X_i + b \right) \geq 1 \ \textit{for}\ i = 1, 2, \cdots, m \tag{9}$$

Through dealing with the optimisation problem below, we can get the optimal hyperplane:

$$\begin{aligned} &\min \ \frac{1}{2} W^T W + C \sum_{i=1}^{m} \varepsilon_i \\ &s.t. \ y_i \left(W \bullet \phi\left(X_i \right) + b \right) \geq 1 - \varepsilon_i, \quad i = 1, 2, \cdots, m, \quad \varepsilon_i \geq 0 \end{aligned} \tag{10}$$

In which C is the penalty parameter, $\varepsilon = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_m\}$ is a slack variable. The penalty parameter C controls the model complexity weigh against classification error. The function $\phi(X_i)$ realises the nonlinear transformation from lower-dimensional samples space to higher-dimensional feature space.

We introduce the non-negative Lagrangian function to tackle the constrained problem. And we select radial basis function (RBF) as kernel function. The RBF is defined as follow:

$$k\left(X_i, X_j \right) = \exp\left(-\gamma \left\| X_i^T - X_j \right\|^2 \right) \quad \gamma > 0 \tag{11}$$

where γ is the span parameter of RBF kernel. The smaller the value is, the wider the kernel spans.

2.4.2 SVM for multiclass classification

To realise the SVM for multi-classification, researchers have proposed many methods. The methods roughly are divided into three categories: one-against-all (OAA), one-against-one (OAO) and all-at-once (AAO). In OAA method, SVM is trained with the positive class representing one class and the negative class representing the others every time. In OAO method, a SVM is trained to classify the i^{th} class and the j^{th} class. Therefore, it produces $n(n - 1)/2$ SVM models. For AAO method, the idea is similar to the OAA, but it can decide n decision functions at once, the i^{th} function splits the i^{th} class from the others.

According to the previous results (Bae et al., 2010), we can know that the training time of OAO method is faster than that of the OAA and AAO method, and the OAO method is more efficient on large scale datasets than OAA method and AAO method. It is

a challenge to deal with the imbalanced multiclass problem. We use the OAO method for multiclass problem and the method is the same to the recognised LIBSVM toolbox.

2.4.3 SVM-weights for multi-classification

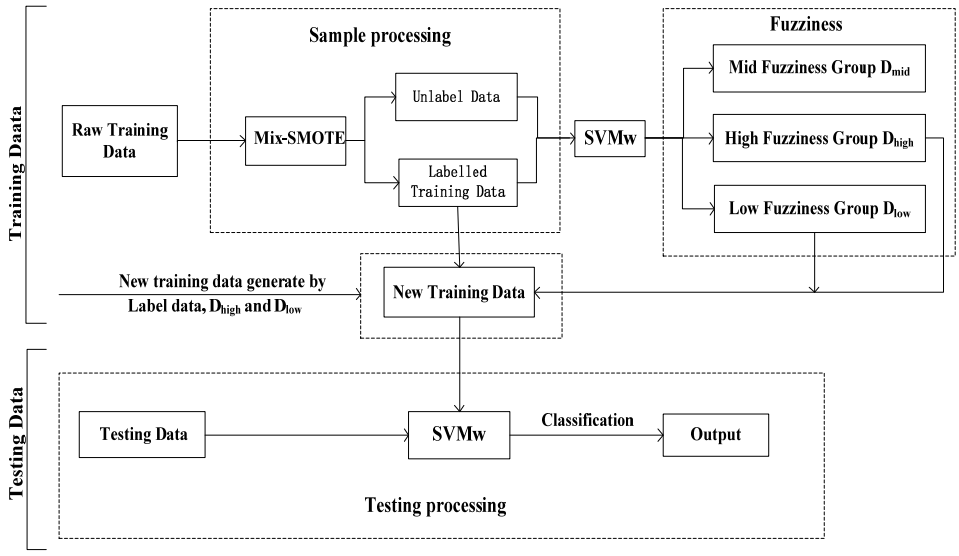
By comparison with SVM, it adds weight vector for each instance when training model, rather than merely consider label vector and feature vector, and can produce better efficiency and higher detection rate compared with the former in the aspect of value misclassification training samples, these samples may play an important role in the final classification decision.

3 Proposed method

On the basis of the partition strategy, we extend this method using unknown label data and proposed a novel algorithm for IDS. We propose the FSVMs algorithm for intrusion detection in this paper. It utilises mix-SMOTE to deal with raw training samples at the beginning, splits the training data by using fuzziness based divide-and-conquer strategy detailed in Table 2. Using the SVMw as basic classifier, we can get accuracy for testing data.

The specific steps of the proposed algorithm FSVMs can be described as Algorithm 1. Firstly, sample training data based on mix-ratio SMOTE. Secondly, training the classifier SVMw, obtaining the membership matrix and fuzziness of unlabelled data, then divide them into three groups: low, mid, high. Thirdly, combine the low fuzziness and high fuzziness data with the raw labelled samples, retrain the classifier model. Finally, obtain the test accuracy in testing dataset. The architecture of developed FSVMs approach is illustrated in Figure 2.

Figure 2 Basic structures of the proposed approach FSVMS for IDS



Algorithm 1 Proposed algorithm FSVMs

Given the training dataset ($tr_{n \times m}$), unlabelled ratio r, testing dataset (te)

1 For raw training data, calculate the number of each class, deciding the sampling ratio for minority classes. Then we can get training data $tr'_{n' \times m} \leftarrow mix\text{-}smote(tr_{n \times m})$.

2 According to the size of dataset, selection of the proper ratio r to split training data: $tr'_{n' \times m} \rightarrow ltr_{n' \times m \times r} + utr_{n' \times m \times (1-r)}$

3 Set the weight vector $(W_{n' \times r \times 1})$ for each instance in ltr.

4 Training classifier: $model = C_{SVMw}(ltr, W)$

5 Obtain the fuzzy membership matrix model $U_{n' \times 5 \times (1-r)} \leftarrow model(utr_{n' \times m \times (1-r)})$

5 Utilising the equation (7) to compute the fuzziness vector: $E_{n' \times (1-r) \times 1} \leftarrow F(U)$

6 According to the magnitude of E, divide unlabelled data into three types: $utr \rightarrow D_{low} + D_{mid} + D_{high}$

7 Add new samples into labelled dataset: and set new weight vector: $nW \leftarrow W + W_{low} + W_{high}$

8 Retraining classifier: $nmodel = C_{SVMw}(nltr, nW)$

9 Obtaining the test accuracy: $te_{acc} \leftarrow nmodel(te)$

4 Dataset and evaluate

In the paper, we use the benchmark intrusion detection dataset and five imbalanced datasets in UCI repository to verify the effectiveness of novel algorithm. Some description and pre-processing are the need to be performed before the experiment. The used dataset finally in two experiments are detailed in Tables 5 and 6.

4.1 Dataset description and pre-processing

4.1.1 Description of KDD Cup'99 dataset

The KDD99 dataset is a subset of the DARPA benchmark dataset. In this paper, we show an overview of the KDD Cup'99 dataset (KDD Cup 1999 Data, 2015), which has been collected by the cyber systems and technology group of MIT Lincoln Laboratory. An analysis on the KDD CUP'99 dataset is found in paper (Tavallaee et al., 2009), some summaries has been made as follows: the original KDD dataset consists of nearly five million labelled records and each record has 41 dimensional features vector. All traffic data belong to normal class or attack class. Attack classes fall into four categories described in Table 3, including kinds of each attack type and corresponding numeric label after the data pre-processing. Each instance is flagged as normal or attack that contains four specific attack types: dos, probe, U2R, R2L.

The KDD Cup'99 dataset's 41 features can be classified into three categories: basic features (F1–F9), content feature (F10–F22), traffic features included time based traffic features (F23–F31) and host based features (F32–F41). The 42 feature is the class label represented various attacks. They can be shown as Table 4 in detail.

Table 3 Different types of attacks in dataset

Attack group	*Attacks*	*Label*
Normal		1
Dos	Back, land, neptune, pod, smurf, teardrop, mailbomb, processtable, udpstorm, apache2,worm	2
Probe	Satan, ipsweep, nmap, portsweep, mscan, saint	3
U2R	Butter_overflow, loadmodule, rootkit, perl, sqlattack, xterm, ps	4
R2L	Guess_passwd, ftp_write, imap, phf, multihop, warezmaster, warezclient, xlock, xsnoop,snmpguess, snmpgetattack, htttptunnel, sendmail, named	5

4.1.2 NSL-KDD dataset

The NLS-KDD Dataset (2015) is the modified version of KDD99 dataset. Though KDD99 and DARPA dataset are widely used in intrusion detection field, these dataset have the huge number of redundant records (Mahoney and Chan, 2003), the issue can affect algorithm's performance for IDS. To solve this issue, researchers have extracted two new datasets: KDD99Train + (NSLtrain) and KDD99Test+ (NSLtest+), which does not include any redundant records in KDD'99.

This dataset has the same features and label with KDD Cup'99, its detail is also presented in Tables 3 and 4.

Table 4 Description of features in KDD Cup'99

Feature no.	*Feature name*	*Feature no.*	*Feature name*	*Feature no.*	*Feature name*
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same srv port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srvserror rate
F12	Loginned in	F26	Srvserror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srvrerror rate
F14	Root shell	F28	Srvrerror rate	F42	Class label

Table 5 The distribution of datasets used experiment1

Dataset	*Training dataset*					*Testing dataset*				
	Normal(%)	*Dos(%)*	*Probe(%)*	*U2R(%)*	*R2L(%)*	*Normal(%)*	*Dos(%)*	*Probe(%)*	*U2R(%)*	*R2L(%)*
Dataset1	19.4	77.44	0.82	0.104	2.245	19.42	73.69	1.335	0.365	5.189
Dataset2	19.77	78.92	0.835	0.021	0.458	19.47	73.85	1.338	0.147	5.201
Dataset3	19.82	79.11	0.837	0.011	0.229	19.48	73.9	1.339	0.073	5.205
Dataset4	19.84	79.2	0.899	0.005	0.115	19.48	73.9	1.339	0.073	5.205
Dataset5	19.85	79.26	0.839	0.002	0.046	19.48	73.9	1.339	0.073	5.205
Dataset6	49.74	33.92	8.61	0.38	7.35	43.08	33.08	10.74	0.89	12.22
Dataset7	51.74	35.28	8.95	0.2	3.82	43.08	33.08	10.74	0.89	12.22
Dataset8	53.02	36.16	9.18	0.08	1.57	18.16	36.64	20.27	1.69	23.24
Dataset9	51.74	35.28	8.95	0.2	3.82	43.08	33.08	10.74	0.89	12.22
Dataset10	53.02	36.16	9.18	0.08	1.57	18.16	36.64	20.27	1.69	23.24

4.1.3 Data pre-processing

Due to the complexity of data, the pre-processing is made before the experiment for acquiring higher accuracy: numerical encoding and normalisation method. Due to the inconsistencies of data range and precision, the attribute data are scaled to the scope of [0, 1]. In addition, we assume that the 1 to 5 represent five classes: 1 for normal, 2 for dos, 3 for probe, 4 for U2R, 5 for R2L.

To meet classifier computational needs that continuous features are used in the model. There are several symbolic features are handled in the same way as continuous features. Many researchers considered different approaches to handle symbolic features, including indicator/dummy variables (Neter et al., 1996), conditional probability (Aha et al., 1991) and clustering technique (Hernández-Pereira et al., 2009) and so on.

In the experiments, the dimensionality of dataset increased from 41 features to 51 by using the indicator variable technique proposed by Neter et al. (1996): if the categorical feature has only two values like (*yes or no*) then it is no requirement to conversion. If distinct values are more than two, it is needed to converse. All the symbolic features are represented into continuous using 1 of *k* coding. In this technique of conversion, 'k' different features are created to represent distinct 'k' values for each categorical feature. For example, if any categorical feature has three values like flag (*S0*, *S1*, *S2*), then it can be transformed into three distinct continuous features like (0, 0, 1), (0, 1, 0) and (1, 0, 0). These '0' and '1' in represented features are considered as decimal number not binary values. If distinct values are not large, this method is more stable than using any single value feature.

Through pre-processing the raw dataset, five complete datasets are used in the paper: KDDtrain, KDDtest, NSLtrain, NSLtest+ and NSLtest–21. In the paper, ten datasets are used for proofing the performance of novel algorithm, in which five subsets are extracted from each complete dataset according to different labels' partition. The detail of these datasets used experiment1 is shown as Table 5. It shows the distribution of every dataset including training and testing set in five classes. The detail of three datasets used experiment3 is shown as Table 6. For each dataset, it includes the number of examples (#Ex.), the number of attribution (#Atts.), the number of classes (#Cl.), the distribution of class (#DC.) and the imbalanced rate (#IR).

Table 6 The distribution of datasets used experiment2

Dataset	*#Ex.*	*#Atts.*	*#Cl.*	*#DC.*	*#IR*
Balance	625	4	3	288/49/288	5.87
Car	1,728	6	4	1210/384/65/69	18.62
Satimage	6,435	36	6	1533/703/1358/626/707/1508	2.45

For tackling imbalanced dataset classification problem, according to different distribution in every dataset, mix ratio is used to sample U2R and R2L in data level.

For the purpose of experiment1, the size of labelled training data is taken much smaller than that of unlabelled data, so that the efficiency of the proposed scheme can be tested properly. The division of training samples and unlabelled samples is based on the ratio of 20:80 through the comparison of several experiments, where 20% is labelled training data, and remaining 80% is unlabelled data. In the experiment2, the datasets are same with experiment1 in terms of distribution and handing. In the experiment3, for each

imbalanced dataset, the division of training dataset and testing dataset is based on the ratio of 10:20, where 10% is training dataset.

4.2 Evaluation criterions

Because KDD Cup'99 is imbalanced datasets, the overall prediction accuracy (OPA) is not effectively used to evaluate classifier. Therefore, accuracy, detection rate, false alarms rate, precision, F-measure and G-mean are computed for the evaluation of imbalanced datasets, which are widely used to evaluate the performance of intrusion detection. They can be calculated by combining a confusion matrix as showed in Table 7.

Table 7 Confusion matrix

↓Actual\Predicted→	*Attacks*	*Normal*
Attacks	TP	FN
Normal	FP	TN

Confusion Matrix is a visualisation tool for comparing the outcome with the ground truth. Each row represents the actual information. Each col represents the prediction value. In the confusion matrix, every instance can be divided into four types: *TP* is the number of true positive, *FP* is the number of false positive, *TN* is the number of true negative, and *FN* is the number of false negative. And positive/negative is interpreted as attack class/normal. The true/false is regarded as right/wrong prediction.

Evaluate metrics used in our research can be calculated by the following formulas (Lin et al., 2015).

Accuracy: in a given test set, the value shows percentage of correct classification tuples by using the classifier. It shows performance of the classification model generally, where the higher accuracy rate, the better the classification model performs.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Precision: refer to the proportion of actual positive samples in predicted positive tuples.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

False alarm rate: it is the proportion of predicted false samples which are normal actually.

$$False\ alarm\ rate = \frac{FP}{TN + FP} \tag{14}$$

Recall: it represents the ratio of actual positive tuples are classified correctly.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

F-measure: it is the harmonic mean of precision and recall rate. It is one of better metrics for imbalanced classification problem. The higher the value represents that the model is more outstanding than others.

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

The G-mean (Sain and Purnami, 2015) is considered as better metric to evaluating performance for imbalanced dataset. Generally, if the minority class is ignored by classifier, it will obtain lower G-mean value. It can be determined as follow:

$$G\text{-}mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} = \sqrt{Sensitivity \times Specificity} \tag{17}$$

Receiver operating characteristic (ROC): is a useful visualisation tool for comparing classification models. It represents the trade-off between true positive rate (TPR) and false positive rate (FPR) values for a given model. TPR's value is the same as Recall/Sensitivity, and FPR is 1-specificity. The closer the upper left corner, the better performance of classifier.

The area under ROC curve (AUC): as the name suggests, it represents the area under the ROC curve. Its value falls in between 0.5 and 1. The higher AUC value, the better performance.

5 Experiment

In this section, three experiments are conduct to verify the performance of the proposed method. All simulations have been run in an Intel® Core™ i7 CPU @ 2.50GHz computer with 4G RAM in Windows 7. The novel algorithm is implemented in MATLAB 2015a and LIBSVM 3.21 version.

In order to verify the effectiveness and robustness of the FSVMs method, this experiment mainly includes three parts: comparing with six basic algorithms for verify the effect of classification, comparing with four algorithms for tackling imbalanced sample problem, comparative experiments on three imbalanced datasets based on SVM classifier from UCI repository are used to confirm the extendibility of proposed algorithm.

5.1 Experiment 1: comparing with six basic algorithms in ten datasets

In the experiment, for verifying the availability and validity of the proposed method, overall accuracy, G-mean value and accuracy for each class in each dataset are compared between FSVMs and six algorithms, the results are represented in Figures 3, 4 and 5. From three histograms, it can be intuitively seen that the proposed method has higher accuracy for each minority class and higher G-mean value for each dataset than others.

In addition, six evaluating metrics are tested in ten datasets for every method, which includes accuracy, recall, false alarm rate, precision, G-mean, F-measures. The results are shown in Tables 8 and 9 (the best result for each column is highlighted in italics). From the tables, it can be discovered that the novel algorithm FSVMs has better performance than the remaining method, the accuracy in three minority classes are far beyond other approaches. For the accuracy in Normal and the Recall, the value is lower because of reducing the ratio of majority class by sampling method. However, proposed method has

higher F-measure and G-mean in each dataset, it can verify that the classifier is more suitable for imbalanced problem.

Figure 3 Comparison of overall accuracy in experiment1 (see online version for colours)

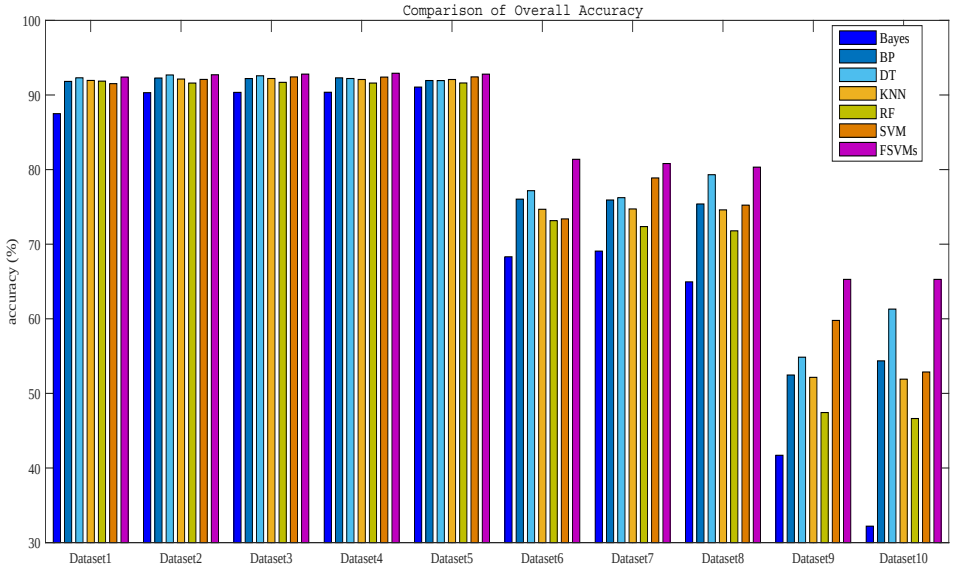
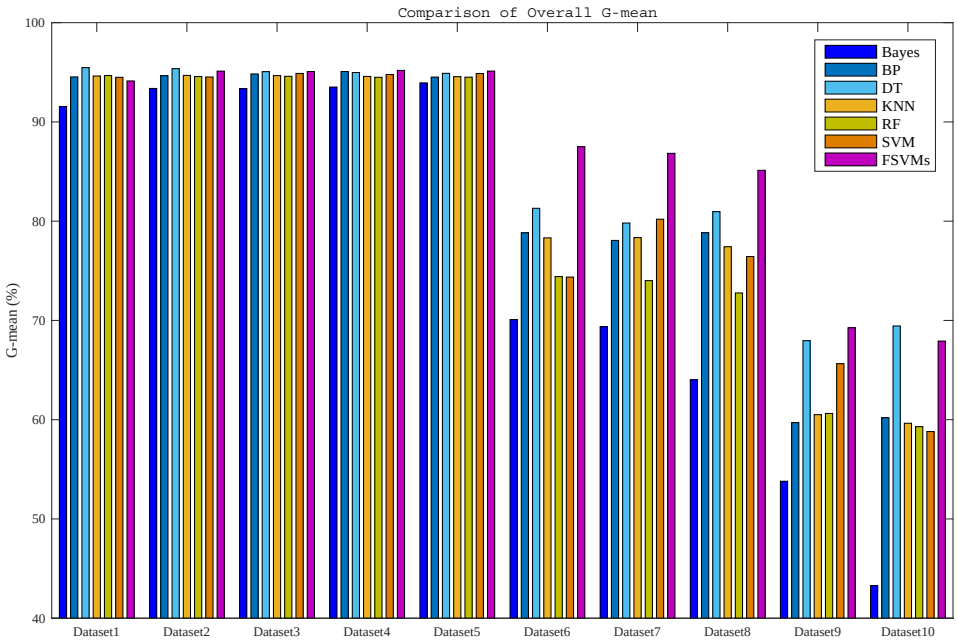


Figure 4 Comparison of overall G-mean in experiment1 (see online version for colours)



The ROC curve is intuitive display for illustrating performance of classifier. The curves corresponding to each classifier in ten datasets are depicted as Figure 6. The closer the upper left corner, the better performance. In the ten figures, even though there are some curves which are close to overlap, the curve of FSVMs is closer to the upper left corner as a whole.

Figure 5 Performance of six methods and FSVMS in each class (see online version for colours)

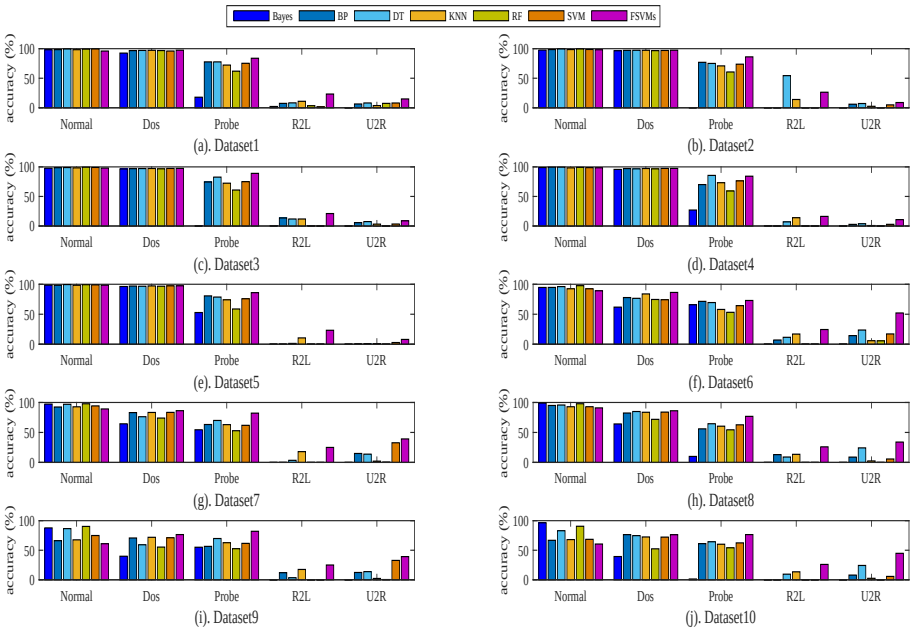


Figure 6 ROC curve comparison in ten datasets (see online version for colours)

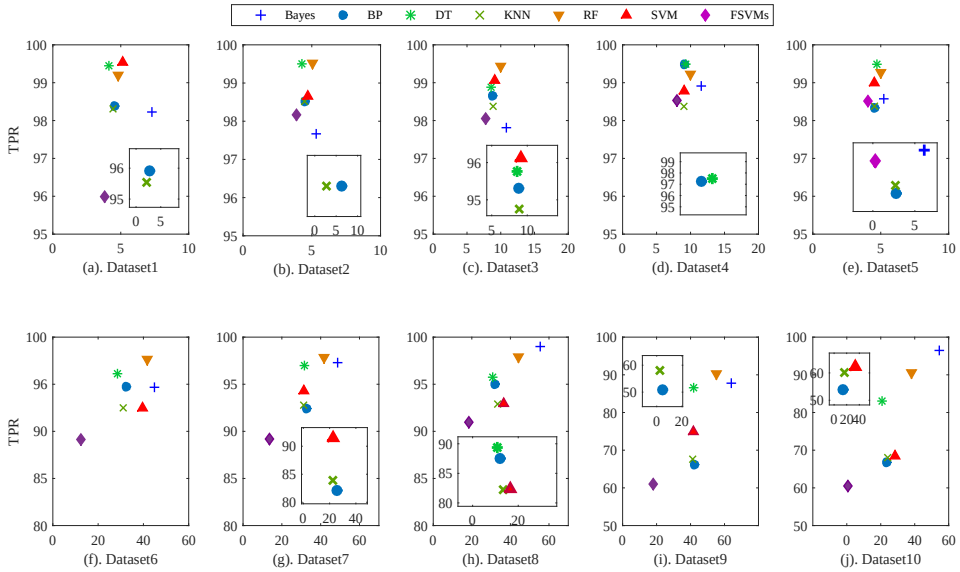


Table 8 Evaluate indicators of five datasets extracted from KDD Cup'99

Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset1	Bayes	98.23±0.02	92.52±0.01	17.89±0.02	2.19±0.02	0	87.49±0.02	99.49±0.01	38.12±0.02	91.88±0.02	91.55±0.02	85.34±0.02
	BP	98.38±0.01	96.77±0.02	77.67±0.02	7.45±0.01	6.43±0.02	91.82±0.01	99.57±0.01	27.72±0.02	95.01±0.01	94.53±0.01	90.84±0.01
	DT	99.44±0.00	97.03±0.01	77.67±0.00	8.33±0.00	8.15±0.01	92.31±0.00	*99.85±0.01*	25.61±0.01	*95.58±0.01*	*95.46±0.01*	91.65±0.01
	KNN	98.31±0.01	97.24±0.01	72.27±0.01	10.96±0.02	3.73±0.01	91.95±0.01	99.55±0.01	27.2±0.01	95.13±0.01	94.62±0.02	91.08±0.00
	RF	99.19±0.02	96.84±0.02	61.94±0.02	3.51±0.01	7.47±0.01	91.86±0.02	99.78±0.02	28.68±0.02	94.84±0.02	94.67±0.01	90.35±0.01
	SVM	*99.54±0.01*	96.02±0.00	75.27±0.01	1.75±0.02	8.09±0.00	91.52±0.02	99.87±0.01	29.99±0.01	94.52±0.01	94.49±0.01	89.71±0.02
	FSVMs	95.99±0.01	*97.41±0.01*	*83.79±0.01*	*23.25±0.01*	*14.95±0.01*	*92.41±0.01*	98.96±0.01	*24.79±0.01*	95.52±0.01	94.13±0.01	*92.31±0.01*
Dataset2	Bayes	97.67±0.02	96.55±0.01	0.05±0.02	0	0	90.31±0.02	99.37±0.01	31.07±0.02	94.05±0.02	93.37±0.02	89.27±0.02
	BP	98.52±0.01	97.16±0.02	76.91±0.02	0	6.04±0.02	92.27±0.01	99.61±0.01	27.48±0.02	95.08±0.01	94.66±0.01	90.96±0.01
	DT	99.51±0.00	97.30±0.01	75.04±0.00	*54.38±0.00*	7.21±0.01	92.68±0.00	99.86±0.01	26.26±0.01	95.45±0.01	95.36±0.01	91.40±0.01
	KNN	98.52±0.01	97.31±0.01	70.91±0.01	14.04±0.02	2.51±0.01	92.14±0.01	99.61±0.01	27.33±0.01	95.11±0.01	94.69±0.02	91.01±0.00
	RF	*99.52±0.02*	96.70±0.02	60.59±0.02	0	0.05±0.01	91.59±0.02	*99.87±0.02*	29.57±0.02	94.61±0.02	94.57±0.01	89.88±0.01
	SVM	98.65±0.01	97.01±0.00	73.74±0.01	0	4.93±0.00	92.09±0.02	99.64±0.01	28.32±0.01	94.89±0.01	94.52±0.01	90.57±0.02
	FSVMs	98.17±0.01	*97.43±0.01*	86.17±0.01	25.44±0.01	*8.93±0.01*	*92.71±0.01*	99.52±0.01	*24.68±0.01*	*95.69±0.01*	*95.12±0.01*	*92.16±0.01*
Dataset3	Bayes	97.81±0.02	96.47±0.01	0.02±0.02	0.43±0.01	0	90.35±0.02	99.41±0.01	31.39±0.02	93.97±0.02	93.36±0.02	89.10±0.02
	BP	98.66±0.01	97.01±0.02	74.48±0.02	13.59±0.01	5.62±0.02	92.20±0.01	99.64±0.01	26.91±0.02	95.21±0.01	94.83±0.01	91.16±0.01
	DT	98.88±0.00	97.17±0.01	82.69±0.00	11.84±0.00	7.44±0.01	92.57±0.00	99.70±0.01	26.38±0.01	95.37±0.01	95.06±0.01	91.39±0.01

Table 8 Evaluate indicators of five datasets extracted from KDD Cup'99 (continued)

Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset3	KNN	98.38±0.01	97.29±0.01	72.42±0.01	11.84±0.02	3.16±0.01	92.21±0.01	99.57±0.01	27.15±0.01	95.14±0.01	94.66±0.02	91.09±0.00
	RF	*99.44±0.02*	96.76±0.02	60.85±0.02	0.44±0.01	0	91.69±0.02	*99.84±0.02*	29.34±0.02	94.67±0.02	94.59±0.01	89.99±0.01
	SVM	99.06±0.01	97.35±0.00	75.01±0.01	0	3.34±0.00	92.42±0.02	99.75±0.01	27.53±0.01	95.11±0.01	94.88±0.01	90.89±0.02
	FSVMs	98.05±0.01	*97.47±0.01*	*89.03±0.01*	*21.05±0.01*	*8.77±0.01*	*92.79±0.01*	99.49±0.01	*24.65±0.01*	*95.69±0.01*	*95.07±0.01*	*92.18±0.01*
Dataset4	Bayes	98.91±0.02	95.71±0.01	27.01±0.02	0.43±0.01	0	90.37±0.02	99.71±0.01	32.61±0.02	93.71±0.02	93.51±0.02	88.41±0.02
	BP	99.49±0.01	97.21±0.02	70.06±0.02	0	2.73±0.02	92.30±0.01	99.86±0.01	27.46±0.02	95.15±0.01	95.07±0.01	90.85±0.01
	DT	*99.49±0.00*	96.72±0.01	*85.59±0.00*	7.02±0.00	3.83±0.01	92.21±0.00	*99.86±0.01*	27.88±0.01	95.05±0.01	94.97±0.01	90.67±0.01
	KNN	98.37±0.01	97.26±0.01	73.24±0.01	14.03±0.02	0.79±0.01	92.07±0.01	99.57±0.01	27.49±0.01	95.06±0.01	94.58±0.02	90.93±0.00
	RF	99.22±0.02	96.72±0.02	59.24±0.02	0	0	91.59±0.02	99.79±0.02	29.36±0.02	94.64±0.02	94.49±0.01	89.99±0.01
	SVM	98.78±0.01	97.41±0.00	76.35±0.01	0	2.81±0.00	92.39±0.02	99.67±0.01	27.48±0.01	95.11±0.01	94.77±0.01	90.93±0.02
	FSVMs	98.53±0.01	*97.45±0.01*	84.23±0.01	*16.23±0.01*	*10.75±0.01*	*92.92±0.01*	99.61±0.01	*25.12±0.01*	*95.64±0.01*	*95.19±0.01*	*91.97±0.01*
Dataset5	Bayes	98.57±0.02	96.27±0.01	52.83±0.02	0	0	91.06±0.02	99.61±0.01	30.46±0.02	94.29±0.02	93.93±0.02	89.51±0.02
	BP	98.33±0.01	97.01±0.02	80.60±0.02	0	0.27±0.02	91.93±0.01	99.56±0.01	27.67±0.02	95.01±0.01	94.52±0.01	90.84±0.01
	DT	*99.48±0.00*	96.69±0.01	78.64±0.00	1.31±0.00	0.72±0.01	91.93±0.00	*99.86±0.01*	28.16±0.01	94.96±0.01	94.89±0.01	90.51±0.01
	KNN	98.37±0.01	97.25±0.01	74.09±0.01	10.52±0.02	0.72±0.01	92.07±0.01	99.57±0.01	27.60±0.01	95.03±0.01	94.56±0.02	90.89±0.00
	RF	99.27±0.02	96.73±0.02	58.76±0.02	0	0	91.61±0.02	99.80±0.02	29.40±0.02	94.63±0.02	94.51±0.01	89.97±0.01
	SVM	98.99±0.01	97.40±0.00	75.85±0.01	0	2.79±0.00	92.42±0.02	99.73±0.01	27.43±0.01	95.13±0.01	94.88±0.01	90.93±0.02
	FSVMs	98.51±0.01	*97.45±0.01*	*85.98±0.01*	*23.25±0.01*	*7.97±0.01*	*92.79±0.01*	99.61±0.01	*25.39±0.01*	*95.57±0.01*	*95.12±0.01*	*91.85±0.01*

Table 9 Evaluating indicators of five datasets extracted from NSL-KDD

Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset6	Bayes	94.68 ± 0.02	61.85 ± 0.01	65.84 ± 0.02	0	0	68.32 ± 0.02	92.31 ± 0.01	38.51 ± 0.02	66.42 ± 0.02	70.08 ± 0.02	51.87 ± 0.02
	BP	94.74 ± 0.01	77.82 ± 0.02	71.46 ± 0.02	7.01 ± 0.01	14.31 ± 0.02	76.04 ± 0.01	93.95 ± 0.01	31.17 ± 0.02	77.25 ± 0.01	78.83 ± 0.01	65.59 ± 0.01
	DT	96.12 ± 0.00	76.53 ± 0.01	69.48 ± 0.00	11.50 ± 0.02	23.64 ± 0.01	77.17 ± 0.00	95.53 ± 0.01	28.18 ± 0.01	79.96 ± 0.01	81.29 ± 0.01	68.76 ± 0.01
	KNN	92.49 ± 0.01	83.87 ± 0.01	57.99 ± 0.01	17.01 ± 0.01	5.85 ± 0.01	74.68 ± 0.01	91.51 ± 0.01	30.75 ± 0.01	76.9 ± 0.01	78.32 ± 0.02	66.32 ± 0.00
	RF	*97.65 ± 0.02*	74.63 ± 0.02	53.21 ± 0.02	0	5.66 ± 0.01	73.15 ± 0.02	*96.85 ± 0.02*	36.07 ± 0.02	71.53 ± 0.02	74.42 ± 0.01	56.72 ± 0.01
	SVM	92.47 ± 0.01	74.23 ± 0.00	64.31 ± 0.01	0	17.14 ± 0.00	73.38 ± 0.02	91.18 ± 0.01	36.15 ± 0.01	72.23 ± 0.01	74.37 ± 0.01	59.81 ± 0.02
	FSVMs	89.14 ± 0.01	*86.35 ± 0.01*	*72.95 ± 0.01*	*24.5 ± 0.01*	*52.11 ± 0.01*	*81.38 ± 0.01*	90.18 ± 0.01	*15.52 ± 0.01*	*87.99 ± 0.01*	*87.51 ± 0.01*	*85.91 ± 0.01*
Dataset7	Bayes	97.29 ± 0.02	64.42 ± 0.01	54.48 ± 0.02	0	0	69.07 ± 0.02	95.88 ± 0.01	39.83 ± 0.02	65.27 ± 0.02	69.38 ± 0.02	49.47 ± 0.02
	BP	92.42 ± 0.01	83.12 ± 0.02	63.24 ± 0.02	0	14.9 ± 0.02	75.92 ± 0.01	91.71 ± 0.01	31.91 ± 0.02	76.72 ± 0.01	78.06 ± 0.01	65.93 ± 0.01
	DT	96.98 ± 0.00	76.25 ± 0.01	70.01 ± 0.00	3.50 ± 0.02	13.7 ± 0.01	76.24 ± 0.00	96.36 ± 0.01	30.12 ± 0.01	78.12 ± 0.01	79.81 ± 0.01	65.68 ± 0.01
	KNN	92.78 ± 0.01	83.35 ± 0.01	63.03 ± 0.01	18.00 ± 0.01	2.1 ± 0.01	74.72 ± 0.01	91.78 ± 0.01	30.79 ± 0.01	76.89 ± 0.01	78.35 ± 0.02	66.16 ± 0.00
	RF	*97.84 ± 0.02*	73.98 ± 0.02	52.91 ± 0.02	0	0.44 ± 0.01	72.36 ± 0.02	*97.02 ± 0.02*	36.05 ± 0.02	70.99 ± 0.02	74.01 ± 0.01	55.98 ± 0.01
	SVM	94.29 ± 0.01	83.46 ± 0.00	61.87 ± 0.01	0	32.78 ± 0.00	78.88 ± 0.02	93.96 ± 0.01	30.51 ± 0.01	79.04 ± 0.01	80.19 ± 0.01	68.21 ± 0.02
	FSVMs	89.19 ± 0.01	*86.32 ± 0.01*	*82.21 ± 0.01*	*25.00 ± 0.01*	*39.18 ± 0.01*	*80.81 ± 0.01*	90.11 ± 0.01	*16.79 ± 0.01*	*87.24 ± 0.01*	*86.84 ± 0.01*	*84.55 ± 0.01*
Dataset8	Bayes	*99.01 ± 0.02*	64.19 ± 0.01	9.99 ± 0.02	0	0	64.95 ± 0.02	*98.11 ± 0.01*	42.54 ± 0.02	58.23 ± 0.02	64.02 ± 0.02	41.41 ± 0.02
	BP	95.01 ± 0.01	82.38 ± 0.02	56.01 ± 0.02	13.00 ± 0.00	8.86 ± 0.02	75.39 ± 0.01	94.12 ± 0.01	30.8 ± 0.02	77.19 ± 0.02	78.84 ± 0.01	65.42 ± 0.01
	DT	95.75 ± 0.00	84.93 ± 0.01	64.56 ± 0.00	9.00 ± 0.02	24.25 ± 0.01	79.32 ± 0.00	95.41 ± 0.01	29.84 ± 0.01	79.72 ± 0.01	80.96 ± 0.01	68.46 ± 0.01
	KNN	92.89 ± 0.01	83.66 ± 0.01	60.43 ± 0.01	13.50 ± 0.01	2.47 ± 0.01	74.61 ± 0.01	91.87 ± 0.01	32.21 ± 0.01	75.81 ± 0.01	77.42 ± 0.02	64.53 ± 0.00
	RF	97.91 ± 0.02	71.86 ± 0.02	54.44 ± 0.02	0	0	71.79 ± 0.02	97.05 ± 0.02	37.35 ± 0.02	69.46 ± 0.02	72.77 ± 0.01	54.08 ± 0.01
	SVM	92.95 ± 0.01	83.92 ± 0.00	62.74 ± 0.01	0	5.70 ± 0.00	75.24 ± 0.02	92.05 ± 0.01	34.18 ± 0.01	74.71 ± 0.01	76.44 ± 0.01	62.87 ± 0.02
	FSVMs	90.97 ± 0.01	*86.19 ± 0.01*	*76.83 ± 0.01*	*26.00 ± 0.01*	*33.95 ± 0.01*	*80.33 ± 0.01*	91.36 ± 0.01	*21.15 ± 0.01*	*85.11 ± 0.01*	*85.12 ± 0.01*	*79.65 ± 0.01*

Table 9 Evaluating indicators of five datasets extracted from NSL-KDD (continued)

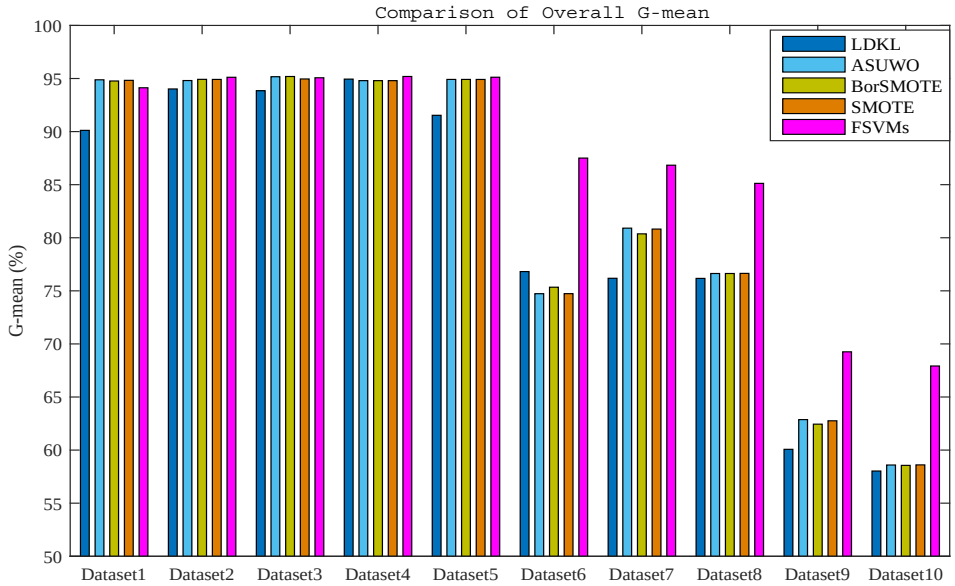
Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset9	Bayes	87.78 ± 0.02	39.84 ± 0.01	55.08 ± 0.02	0	0	41.71 ± 0.02	92.07 ± 0.01	76.66 ± 0.02	48.55 ± 0.02	53.79 ± 0.02	32.97 ± 0.02
	BP	66.17 ± 0.01	70.70 ± 0.02	56.53 ± 0.02	12.01 ± 0.00	12.38 ± 0.02	52.46 ± 0.01	86.81 ± 0.01	74.24 ± 0.02	66.48 ± 0.02	59.71 ± 0.01	53.87 ± 0.01
	DT	86.57 ± 0.00	59.25 ± 0.01	69.81 ± 0.00	3.50 ± 0.02	13.79 ± 0.01	54.85 ± 0.00	94.13 ± 0.01	68.52 ± 0.01	68.09 ± 0.01	67.95 ± 0.01	53.34 ± 0.01
	KNN	67.61 ± 0.01	71.87 ± 0.01	62.73 ± 0.01	17.50 ± 0.01	2.25 ± 0.01	52.15 ± 0.01	87.14 ± 0.01	73.33 ± 0.01	66.79 ± 0.01	60.50 ± 0.02	54.14 ± 0.00
	RF	*90.28 ± 0.02*	55.32 ± 0.02	52.62 ± 0.02	0	0.43 ± 0.01	47.43 ± 0.02	*94.62 ± 0.01*	73.37 ± 0.02	56.93 ± 0.02	60.63 ± 0.01	40.71 ± 0.01
	SVM	74.91 ± 0.01	71.14 ± 0.00	61.61 ± 0.01	0	32.78 ± 0.00	59.78 ± 0.02	91.01 ± 0.02	71.48 ± 0.01	70.49 ± 0.01	65.64 ± 0.01	57.52 ± 0.02
	FSVMs	61.01 ± 0.01	*76.51 ± 0.01*	*82.06 ± 0.01*	*25.01 ± 0.01*	*39.22 ± 0.01*	*66.28 ± 0.01*	88.45 ± 0.01	*57.09 ± 0.01*	*83.24 ± 0.01*	*69.26 ± 0.01*	*78.62 ± 0.01*
Dataset10	Bayes	*96.46 ± 0.02*	39.33 ± 0.01	1.33 ± 0.02	0	0	32.20 ± 0.02	*95.81 ± 0.01*	77.67 ± 0.02	32.29 ± 0.02	43.27 ± 0.02	19.41 ± 0.02
	BP	66.77 ± 0.01	*76.39 ± 0.02*	61.16 ± 0.02	0	7.95 ± 0.02	54.36 ± 0.01	87.50 ± 0.01	74.59 ± 0.02	66.98 ± 0.02	60.19 ± 0.01	54.26 ± 0.01
	DT	83.04 ± 0.00	74.71 ± 0.01	64.36 ± 0.00	9.50 ± 0.02	24.26 ± 0.00	61.29 ± 0.00	93.75 ± 0.01	68.87 ± 0.01	71.72 ± 0.01	*69.44 ± 0.01*	58.07 ± 0.01
	KNN	68.03 ± 0.01	72.46 ± 0.01	60.16 ± 0.01	13.50 ± 0.01	2.50 ± 0.01	51.91 ± 0.01	87.20 ± 0.01	74.49 ± 0.01	65.37 ± 0.01	59.64 ± 0.02	52.28 ± 0.00
	RF	90.56 ± 0.02	52.41 ± 0.02	54.16 ± 0.02	0	0	46.63 ± 0.02	94.63 ± 0.01	74.3 ± 0.02	55.06 ± 0.02	59.29 ± 0.01	38.82 ± 0.01
	SVM	68.44 ± 0.01	72.21 ± 0.00	62.44 ± 0.01	0	5.70 ± 0.01	52.87 ± 0.02	87.59 ± 0.02	76.11 ± 0.01	64.07 ± 0.01	58.8 ± 0.01	50.52 ± 0.02
	FSVMs	60.51 ± 0.01	76.28 ± 0.01	*76.48 ± 0.01*	*26.01 ± 0.01*	*44.77 ± 0.01*	*65.28 ± 0.01*	88.33 ± 0.01	*60.62 ± 0.01*	*81.85 ± 0.01*	67.92 ± 0.01	*76.25 ± 0.01*

Table 10 The AUC value obtained for different methods

Dataset	*Model*						
	Bayes	*BP*	*DT*	*KNN*	*RF*	*SVM*	*FSVMs*
Dataset1	0.839	0.8943	*0.9124*	0.896	0.8966	0.893	0.8867
Dataset2	0.8727	0.8963	*0.9098*	0.897	0.8947	0.8936	0.9055
Dataset3	0.9421	0.9543	0.9541	0.9528	0.9505	*0.9543*	0.9536
Dataset4	0.9435	*0.9557*	0.9523	0.9524	0.9494	0.9535	0.9549
Dataset5	0.9465	0.9517	0.9523	0.9521	0.9496	0.9539	*0.9541*
Dataset6	0.7512	0.8122	0.8389	0.8073	0.7807	0.7655	*0.8834*
Dataset7	0.7446	0.7979	0.8279	0.8078	0.7815	0.8154	*0.8782*
Dataset8	0.7106	0.8146	0.8255	0.7965	0.7703	0.7828	*0.8632*
Dataset9	0.6163	0.6179	*0.7215*	0.6309	0.6731	0.6637	0.7141
Dataset10	0.6072	0.6147	*0.7101*	0.6186	0.6594	0.5983	0.6976
Average	0.7974	0.8309	0.8605	0.8311	0.8306	0.8374	*0.8691*

The AUC value analysis also is used in this comparative study. It is the numerical description for performance shown in Figure 6. Table 10 shows the result of AUC, all values are listed for each dataset by using FSVMs and other six learning algorithms the mean value was computed for each dataset by applying methods. From Table 10, it is clear that the proposed method has higher value about 4% averagely in the existing ten datasets.

Figure 7 Comparison of overall G-mean in experiment2 (see online version for colours)

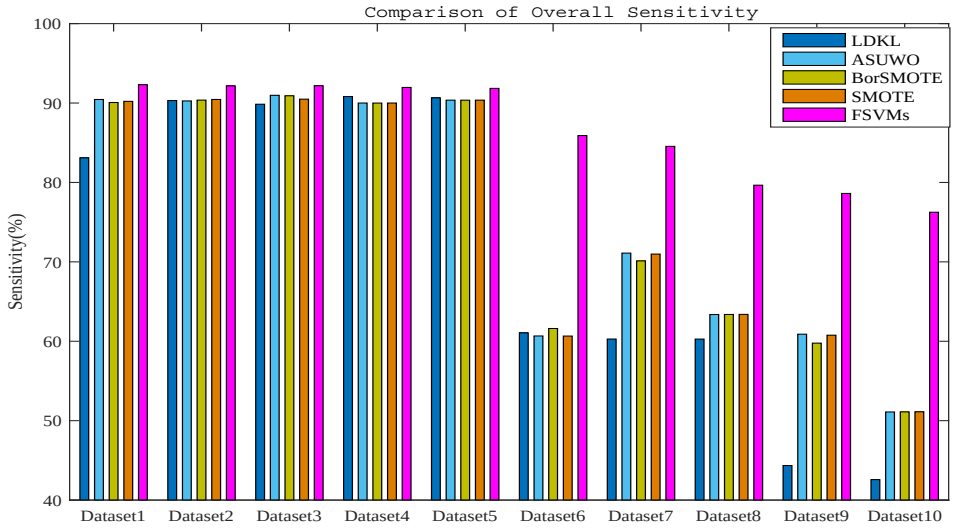


5.2 Experiment 2 comparing with four algorithms for imbalanced problem

For verify the effectiveness of proposed method, the comparison between four algorithms and FSVMs is executed in the experiment, which including LDKL, ASUWO, BorSMOTE, SMOTE. The results are shown in Table 11 (the best result for each column is highlighted in italics). From Table 11, it can be discovered that the accuracy in three minority classes are far beyond others, and proposed method has higher F-measure and G-mean, which are more suitable metrics for imbalanced classification problem.

In addition, Figure 7 presents the comparison of overall G-mean between FSVMs and four algorithms in each dataset. The histograms of sensitivity for each dataset are showed in Figure 8. From two figures, the value of proposed method (purple in the picture) is higher than others. So, it can verify that the effectiveness of proposed method in solving the imbalanced problem.

Figure 8 Comparison of overall sensitivity in experiment2 (see online version for colours)



5.3 Experiment 3 in three imbalanced datasets based on SVM classifier

Comparative experiments on three imbalanced datasets from UCI repository are used to confirm the extendibility of the proposed algorithm. The evaluated results are shown in Table 12 (the best result for each column is highlighted in italics), from the table, proposed approach has higher value in six metrics medially, it can be verified that the novel method is very suitable for processing the imbalanced dataset in most multi-classification cases.

Table 11 Evaluating Indicators of five algorithms in ten IDS dataset

Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset1	LDKL	97.72±0.02	88.99±0.02	57.51±0.02	3.51±0.02	1.57±0.03	85.42±0.02	99.33±0.01	41.57±0.01	90.49±0.02	90.11±0.02	83.10±0.02
	ASUWO	99.52±0.01	96.78±0.01	49.22±0.02	13.59±0.01	9.88±0.01	91.87±0.01	99.87±0.01	28.37±0.02	94.92±0.01	*94.88±0.01*	90.44±0.01
	BorSMOTE	*99.71±0.01*	96.78±0.01	54.02±0.01	5.70±0.03	4.51±0.01	91.65±0.01	*99.92±0.01*	29.16±0.01	94.73±0.01	94.76±0.01	90.06±0.01
	SMOTE	99.67±0.02	96.78±0.02	60.74±0.01	15.35±0.01	4.51±0.02	91.78±0.02	99.91±0.01	28.86±0.02	94.81±0.01	94.82±0.01	90.21±0.02
	FSVMs	95.99±0.01	*97.41±0.01*	*83.79±0.01*	*23.25±0.01*	*14.95±0.01*	*92.41±0.01*	98.96±0.02	*24.79±0.01*	*95.52±0.01*	94.13±0.02	*92.31±0.01*
Dataset2	LDKL	97.87±0.02	96.91±0.02	47.91±0.03	0	0.13±0.03	91.27±0.02	99.43±0.01	28.91±0.01	94.65±0.02	94.01±0.02	90.31±0.02
	ASUWO	99.58±0.01	96.61±0.01	62.69±0.01	15.78±0.01	3.96±0.01	91.81±0.01	99.88±0.01	28.72±0.02	94.83±0.01	94.80±0.01	90.26±0.01
	BorSMOTE	*99.7±0.01*	96.61±0.01	57.51±0.02	4.82±0.03	3.27±0.01	91.71±0.01	*99.92±0.01*	28.42±0.01	94.90±0.01	94.92±0.01	90.36±0.01
	SMOTE	99.6±0.02	96.62±0.02	57.51±0.01	19.29±0.01	5.71±0.02	91.83±0.02	99.89±0.01	28.29±0.02	94.93±0.01	94.91±0.01	90.45±0.02
	FSVMs	98.17±0.02	*97.43±0.01*	*86.17±0.01*	*25.44±0.01*	*8.93±0.01*	*92.71±0.01*	99.52±0.02	*24.68±0.01*	*95.69±0.01*	*95.12±0.01*	*92.16±0.01*
Dataset3	LDKL	98.02±0.02	92.4±0.02	45.07±0.02	0	0.48±0.02	88.01±0.02	99.44±0.02	28.95±0.01	94.4±0.02	93.85±0.02	89.85±0.02
	ASUWO	99.55±0.01	96.63±0.01	65.65±0.01	17.54±0.01	8.78±0.01	92.15±0.01	99.88±0.01	27.12±0.02	95.21±0.01	95.16±0.01	90.97±0.01
	BorSMOTE	*99.66±0.01*	96.63±0.02	62.77±0.02	21.92±0.02	5.97±0.02	91.99±0.02	*99.91±0.01*	27.19±0.02	95.20±0.01	*95.19±0.01*	90.91±0.01
	SMOTE	99.62±0.02	96.64±0.01	62.79±0.01	17.54±0.01	7.58±0.01	92.07±0.01	99.90±0.01	28.24±0.01	94.96±0.02	94.95±0.02	90.48±0.02
	FSVMs	98.05±0.02	*97.47±0.01*	*89.03±0.01*	*21.05±0.01*	*8.77±0.01*	*92.79±0.01*	99.49±0.02	*24.65±0.01*	*95.69±0.01*	95.07±0.02	*92.18±0.01*
Dataset4	LDKL	99.25±0.02	96.97±0.02	42.33±0.02	0	1.85±0.02	91.65±0.02	99.8±0.02	27.44±0.02	95.09±0.01	94.94±0.02	90.81±0.02
	ASUWO	99.85±0.01	96.67±0.01	64.40±0.01	8.33±0.01	0	91.76±0.01	99.96±0.00	29.22±0.01	94.72±0.01	94.79±0.01	90.01±0.02
	BorSMOTE	*99.85±0.01*	96.68±0.02	64.41±0.01	3.07±0.02	0	91.76±0.02	*99.97+0.01*	29.24±0.01	94.71±0.02	94.80±0.01	90.89±0.01
	SMOTE	99.85±0.02	96.67±0.01	64.40±0.02	7.02±0.01	0	91.77±0.01	99.96±0.01	29.23±0.01	94.72±0.02	94.79±0.02	90.01±0.01
	FSVMs	98.53±0.02	*97.45±0.01*	84.23±0.01	*16.23±0.01*	*10.75±0.01*	*92.92±0.01*	99.61±0.02	*25.12±0.01*	*95.64±0.01*	*95.19±0.01*	*91.97±0.01*
Dataset5	LDKL	92.42±0.02	96.92±0.01	56.24±0.02	0	0.01±0.03	90.39±0.02	98.01±0.02	29.29±0.01	94.18±0.02	91.53±0.02	90.65±0.02
	ASUWO	99.69±0.01	96.98±0.02	71.77+0.00	5.26±0.01	0	92.06±0.01	99.91±0.01	28.51±0.01	94.90±0.01	94.91±0.01	90.36±0.01
	BorSMOTE	*99.70±0.01*	96.98±0.01	71.77±0.01	0.44±0.02	0	92.05±0.01	*99.92±0.01*	28.52±0.01	94.89±0.01	94.92±0.01	90.36±0.00
	SMOTE	99.68±0.02	96.99±0.01	71.78±0.01	6.57±0.01	0	92.06±0.01	99.91±0.02	28.51±0.02	94.90±0.02	94.91±0.02	90.36±0.02
	FSVMs	98.51±0.02	*97.45±0.01*	*85.98±0.01*	*23.25±0.01*	*7.97±0.01*	*92.79±0.01*	99.61±0.01	*25.39±0.01*	*95.57±0.01*	*95.12±0.01*	*91.85±0.01*

Table 11 Evaluating Indicators of five algorithms in ten IDS dataset (continued)

Dataset	*Model*	*Normal*	*DOS*	*Probe*	*U2R*	*R2l*	*Acc*	*Recall*	*FAR*	*F-measure*	*G-mean*	*Precision*
Dataset6	LDKL	*96.64±0.02*	73.49±0.02	41.74±0.02	0.50±0.00	11.18±0.02	71.81±0.02	*95.42±0.01*	31.61±0.02	74.46±0.01	76.82±0.01	61.05±0.01
	ASUWO	92.05±0.02	74.32±0.02	64.43±0.01	25.50±0.00	17.02±0.01	73.46±0.01	90.81±0.01	35.60±0.01	72.73±0.02	74.72±0.01	60.66±0.02
	BorSMOTE	92.14±0.01	74.32±0.01	64.31±0.01	22.01±0.02	17.06±0.01	73.47±0.01	90.89±0.01	34.65±0.01	73.43±0.01	75.34±0.02	61.61±0.01
	SMOTE	92.08±0.01	74.32±0.02	64.31±0.02	*28.01±0.01*	17.02±0.02	73.48±0.02	90.83±0.02	35.61±0.01	72.74±0.02	74.73±0.02	60.65±0.02
	FSVMs	89.14±0.02	*86.35±0.01*	*72.95±0.01*	24.50±0.02	*52.11±0.01*	*81.38±0.01*	90.18±0.02	*15.52±0.01*	*87.99±0.01*	*87.51±0.01*	*85.91±0.01*
Dataset7	LDKL	*96.28±0.01*	73.92±0.02	42.37±0.02	0	12.59±0.02	72.01±0.02	*95.01±0.01*	32.67±0.02	73.76±0.02	76.18±0.02	60.28±0.02
	ASUWO	92.06±0.02	83.57±0.01	61.91±0.01	23.01±0.01	32.66±0.02	78.15±0.01	91.84±0.02	28.29±0.01	80.15±0.01	80.90±0.01	71.10±0.01
	BorSMOTE	92.11±0.01	83.58±0.01	61.87±0.01	17.50±0.01	32.71±0.01	78.13±0.01	91.88±0.01	29.22±0.01	79.54±0.01	80.36±0.01	70.12±0.02
	SMOTE	92.01±0.02	83.58±0.02	61.86±0.02	18.50±0.02	32.67±0.01	78.09±0.02	91.79±0.01	28.39±0.02	80.05±0.02	80.81±0.02	70.98±0.01
	FSVMs	89.19±0.02	*86.32±0.01*	*82.21±0.01*	*25.01±0.01*	*39.18±0.01*	*80.81±0.01*	90.11±0.02	*16.79±0.01*	*87.24±0.01*	*86.84±0.01*	*84.55±0.01*
Dataset8	LDKL	96.26±0.02	74.91±0.02	40.11±0.02	3.01±0.02	1.48±0.02	70.76±0.02	*94.79±0.01*	31.77±0.02	73.69±0.02	76.17±0.02	60.27±0.02
	ASUWO	92.67±0.01	83.93±0.01	62.74±0.01	13.50±0.01	5.62±0.01	75.23±0.01	91.80±0.01	33.83±0.01	74.98±0.01	76.63±0.01	63.36±0.01
	BorSMOTE	92.66±0.01	83.93±0.01	62.74±0.01	13.50±0.02	5.70±0.02	75.24±0.01	91.79±0.01	33.84±0.02	74.98±0.02	76.63±0.02	63.37±0.01
	SMOTE	*92.68±0.01*	83.93±0.02	62.74±0.02	14.50±0.01	5.70±0.01	75.25±0.02	91.80±0.02	33.82±0.01	74.99±0.01	76.64±0.01	63.38±0.02
	FSVMs	90.97±0.02	*86.19±0.01*	*76.83±0.01*	*26.01±0.01*	*33.95±0.01*	*80.33±0.01*	91.36±0.02	*21.15±0.01*	*85.11±0.01*	*85.12±0.01*	*79.65±0.01*
Dataset9	LDKL	*81.36±0.01*	52.21±0.02	63.86±0.01	2.50±0.02	0.11±0.02	46.91±0.02	*90.47±0.01*	73.18±0.01	59.52±0.02	60.07±0.02	44.35±0.02
	ASUWO	64.91±0.02	71.30±0.02	61.65±0.01	23.5±0.01	32.67±0.01	58.41±0.01	87.97±0.02	71.74±0.02	71.96±0.01	62.87±0.01	60.89±0.01
	BorSMOTE	65.24±0.01	71.32±0.01	61.62±0.02	18.01±0.02	32.71±0.02	58.38±0.01	88.05±0.01	72.55±0.01	71.20±0.02	62.44±0.02	59.76±0.02
	SMOTE	64.82±0.01	71.32±0.01	61.61±0.02	19.01±0.01	32.67±0.01	58.31±0.02	87.93±0.01	71.85±0.02	71.86±0.01	62.76±0.01	60.76±0.01
	FSVMs	61.01±0.02	*76.51±0.01*	*82.06±0.01*	*25.01±0.01*	*39.22±0.01*	*66.28±0.01*	88.45±0.02	*57.09±0.01*	*83.24±0.01*	*69.26±0.01*	*78.62±0.01*
Dataset10	LDKL	*79.08±0.01*	48.11±0.02	59.07±0.02	0	0.07±0.02	43.98±0.02	*88.63±0.01*	73.55±0.02	57.52±0.02	58.02±0.02	42.57±0.02
	ASUWO	67.19±0.02	72.22±0.01	62.44±0.01	13.01±0.02	5.62±0.01	52.85±0.01	87.21±0.01	76.12±0.01	64.44±0.02	58.59±0.01	51.09±0.02
	BorSMOTE	67.10±0.01	72.21±0.01	62.45±0.01	13.00±0.01	5.71±0.01	52.86±0.02	87.19±0.02	76.14±0.01	64.46±0.01	58.56±0.02	51.11±0.01
	SMOTE	67.19±0.02	72.22±0.02	62.45±0.02	14.00±0.02	5.63±0.02	52.87±0.01	87.22±0.01	76.11±0.02	64.45±0.01	58.61±0.01	51.12±0.02
	FSVMs	60.51±0.02	*76.28±0.01*	*76.48±0.01*	*26.01±0.01*	*44.77±0.01*	*65.28±0.01*	88.33±0.02	*60.62±0.01*	*81.85±0.01*	*67.92±0.01*	*76.25±0.01*

Figure 9 Comparison of overall G-mean in experiment3 (see online version for colours)

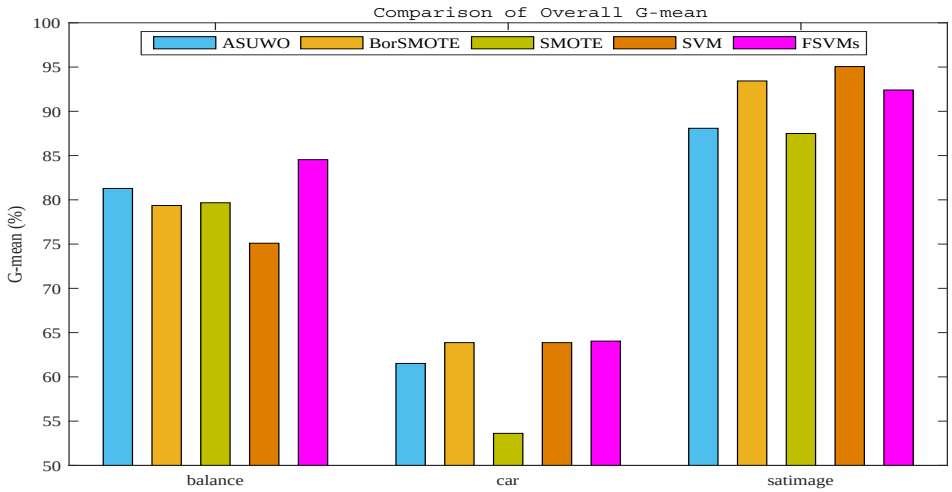


Figure 9 presents the comparison of overall G-mean between FSVMs and four algorithms based on SVM classifier in each dataset, including ASUWO, BorSMOTE, SMOTE and SVM. The histograms of Sensitivity for each dataset are showed in Figure 10. From these figures, the value of proposed method (purple in the picture) is higher, which represents the effectiveness of approach in solving imbalanced dataset

Figure 10 Comparison of overall sensitivity in experiment3 (see online version for colours)

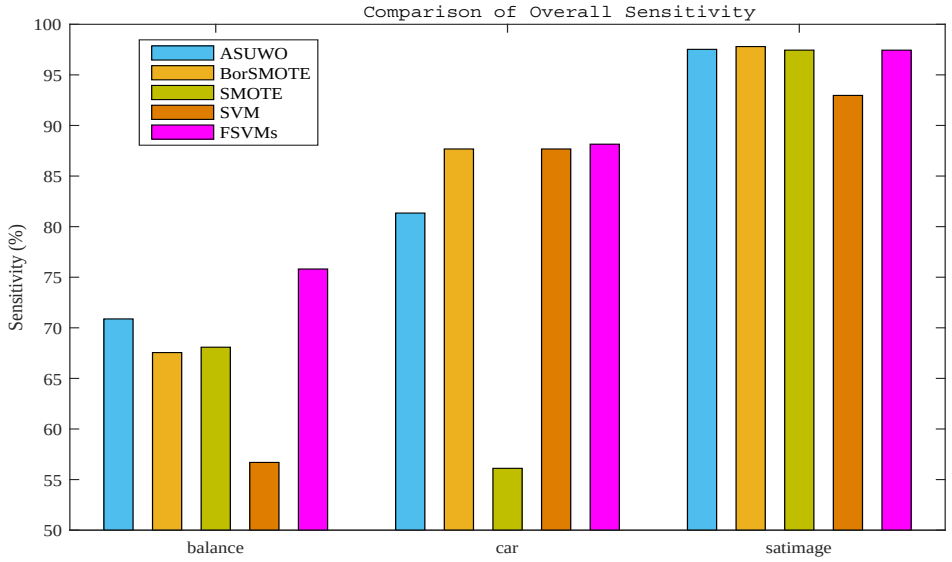


Table 12 Evaluating indicators of three imbalanced datasets

Dataset	*Model*	*Acc*	*Recall*	*FAR*	*F-measure*	*Precision*	*G-mean*
Balance	ASUWO	74.03±0.01	90.84±0.01	22.84±0.01	79.62±0.01	70.87±0.02	81.28±0.01
	BorSMOTE	73.55±0.02	90.71±0.02	25.41±0.02	77.43±0.02	67.55±0.02	79.35±0.02
	SMOTE	73.79±0.01	90.78±0.02	25.10±0.02	77.81±0.02	68.08±0.01	79.67±0.01
	SVM	76.44±0.01	*99.21±0.01*	33.68±0.01	72.15±0.01	56.69±0.02	75.10±0.02
	FSVMs	*82.69±0.01*	93.68±0.01	*22.32±0.01*	*83.80±0.01*	*75.81±0.01*	*84.54±0.01*
Car	ASUWO	42.05±0.02	20.18±0.02	6.25±0.02	32.34±0.01	81.34±0.01	61.51±0.02
	BorSMOTE	48.65±0.01	30.03±0.01	6.48±0.01	44.74±0.02	87.67±0.02	63.86±0.01
	SMOTE	48.74±0.01	30.14±0.02	6.25±0.01	*44.92±0.02*	56.11±0.01	53.62±0.02
	SVM	48.64±0.02	30.03±0.01	6.48±0.02	44.74±0.01	87.67±0.01	63.86±0.01
	FSVMs	*51.43±0.02*	*31.29±0.01*	*5.31±0.01*	40.17±0.01	*88.15±0.01*	*64.04±0.01*
Satimage	ASUWO	76.20±0.01	89.54±0.02	5.29±0.01	93.36±0.02	97.52±0.01	88.08±0.02
	BorSMOTE	69.20±0.02	93.67±0.01	3.85±0.02	*95.69±0.02*	*97.79±0.02*	93.43±0.01
	SMOTE	70.50±0.02	89.95±0.02	6.02±0.01	93.55±0.02	97.44±0.01	87.49±0.02
	SVM	70.05±0.02	98.14±0.01	10.40±0.02	95.48±0.01	92.97±0.02	*95.05±0.01*
	FSVMs	*82.25±0.01*	*93.90±0.01*	*5.38±0.01*	95.63±0.01	97.45±0.01	92.41±0.01

6 Conclusions and future research

In this paper, a novel efficient approach FSVMs is proposed for detection network intrusion. In which, a mix-SMOTE sampling technique is introduced to pre-process the imbalanced dataset. Then, fuzziness technology based SVM classifier is combined to effectively identify network traffic. Through experiments in three aspect, it is verified that the proposed algorithm is not only outperformed especially for minority classes (R2L and U2R) attacks in terms of detection rate and stability in the field of network intrusion detection, but also has a better performance than traditional classification engines on imbalanced datasets.

However, the proposed method obtained the higher accuracy for imbalance datasets, it still has some limitations. Firstly, when it obtains the higher accuracy for minority samples, the detection rate for majority classes is lower than that of other classifiers. Secondly, the method's effect is restricted by the imbalanced rate of dataset. Therefore, in the future research, the focus is to improve the recognition rate of the majority while maintaining the higher accuracy of the minority. Another goal is to explore a more suitable mechanism to sample in face of the higher unbalanced rate.

Acknowledgements

This work was supported by the Top discipline construction for Senior School in Ningxia of China (Grant No. NXYLXK2017B11) and the Key Research Fund of Ningxia Normal University of China (Grant No. NXSFZDA1801 and NXSFZDA1802).

References

Aburomman, A.A. and Ibne Reaz, M.B. (2016) 'A novel SVM-kNN-PSO ensemble method for intrusion detection system', *Applied Soft Computing*, Vol. 38, pp.360–372.

Aha, D.W. et al. (1991) 'Instance-based learning algorithms', *Machine Learning*, Vol. 6, No. 1, pp.37–66.

Ahmed, M. et al. (2016) 'A survey of network anomaly detection techniques', *Journal of Network and Computer Applications*, Vol. 60, pp.19–31.

Ashfaq, R.A.R. et al. (2016) 'Fuzziness based semi-supervised learning approach for intrusion detection system', *Information Sciences*, Vol. 378, pp.484–497.

Bae, M.H. et al. (2010) 'Mix-ratio sampling: classifying multiclass imbalanced mouse brain images using support vector machine', *Expert Systems with Applications*, Vol. 37, No. 7, pp.4955–4965.

Bhuyan, M.H. et al. (2016) 'A multi-step outlier-based anomaly detection approach to network-wide traffic', *Information Sciences*, Vol. 348, pp.243–271.

Canbay, Y. and Sagiroglu, S. (2015) 'A hybrid method for intrusion detection', in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp.156–161.

Chawla, N.V. et al. (2002) 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, Vol. 16, pp.321–357.

Cijo, J. et al. (2013) 'Local deep kernel learning for efficient non-linear SVM prediction', in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, pp.286–494.

De La Hoz, E. et al. (2014) 'Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps', *Knowledge-Based Systems*, Vol. 71, pp.322–338.

De La Hoz, E. et al. (2015) 'PCA filtering and probabilistic SOM for network intrusion detection', *Neurocomputing*, Vol. 164, pp.71–81.

De Luca, A. and Termini, S. (1972) 'A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory', *Information and Control*, Vol. 20, No. 4, pp.301–312.

Devarakonda, N. et al. (2012) 'Intrusion detection system using bayesian network and hidden markov model', *Procedia Technology*, Vol. 4, pp.506–514.

Eesa, A.S. et al. (2015) 'A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems', *Expert Systems with Applications*, Vol. 42, No. 5, pp.2670–2679.

Feng, W. et al. (2014) 'Mining network data for intrusion detection through combining SVMs with ant colony networks', *Future Generation Computer Systems*, Vol. 37, No. 7, pp.127–140.

Fossaceca, J.M. et al. (2015) 'MARK-ELM: application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection', *Expert Systems with Applications*, Vol. 42, No. 8, pp.4062–4080.

García, S. et al. (2012) 'Evolutionary-based selection of generalized instances for imbalanced classification', *Knowledge-Based Systems*, Vol. 25, No. 1, pp.3–12.

García, S. et al. (2016) 'Tutorial on practical tips of the most influential data preprocessing algorithms in data mining', *Knowledge-Based Systems*, Vol. 98, pp.1–29.

Hamed, K. and Hamid, M. (2014) 'An intelligent intrusion detection system based on expectation maximization algorithm in wireless sensor networks', *International Journal of Information and Communication Technology Research*, Vol. 4, No. 1, pp.1–10.

Hasan, M.A.M. et al. (2014) 'Support vector machine and random forest modeling for intrusion detection system (IDS)', *Journal of Intelligent Learning Systems and Applications*, Vol. 6, No. 1, pp.45–52.

Hernández-Pereira, E. et al. (2009) 'Conversion methods for symbolic features: a comparison applied to an intrusion detection problem', *Expert Systems with Applications*, Vol. 36, No. 7, pp.10612–10617.

Hosseini Bamakan, S.M. et al. (2016) 'An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization', *Neurocomputing*, Vol. 199, pp.90–102.

Huang, J-Y. et al. (2013) 'Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining', *Information Sciences*, Vol. 231, pp.32–44.

Ji, S-Y. et al. (2016) 'A multi-level intrusion detection method for abnormal network behaviors', *Journal of Network and Computer Applications*, Vol. 62, pp.9–17.

Jian, C. et al. (2016) 'A new sampling method for classifying imbalanced data based on support vector machine ensemble', *Neurocomputing*, Vol. 193, pp.115–122.

Karami, A. and Guerrero-Zapata, M. (2015) 'A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks', *Neurocomputing*, Vol. 149, pp.1253–1269.

KDD Cup 1999 Data (2015) [online] *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html* (accessed 23 June 2015).

Kim, G. et al. (2014) 'A novel hybrid intrusion detection method integrating anomaly detection with misuse detection', *Expert Systems with Applications*, Vol. 41, No. 4, pp.1690–1700.

Koc, L. et al. (2012) 'A network intrusion detection system based on a hidden naïve Bayes multiclass classifier', *Expert Systems with Applications*, Vol. 39, No. 18, pp.13492–13500.

Lin, W-C. et al. (2015) 'CANN: an intrusion detection system based on combining cluster centers and nearest neighbors', *Knowledge-Based Systems*, Vol. 78, No. 1, pp.13–21.

Mahoney, M.V. and Chan, P.K. (2003) *An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data For Network Anomaly Detection*, in Vigna, G. et al. (Eds.), pp.220–237, Springer Berlin Heidelberg, Berlin, Heidelberg.

Nekooeimehr, I. and Lai-Yuen, S.K. (2016) 'Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets', *Expert Systems with Applications*, Vol. 46, pp.405–416.

Neter, J. et al. (1996) *Applied Linear Statistical Models*, lrwin, Chicago.

NSL-KDD Dataset (2015) [online] *http://nsl.cs.unb.ca/NSL-KDD/* (accessed 23 June 2015).

Ojala, J. (2013) 'Personal content in online sports communities: motivations to capture and share personal exercise data', *International Journal of Social and Humanistic Computing*, Vol. 2, Nos. 1–2, pp.68–85.

Ravale, U. et al. (2015) 'Feature selection based hybrid anomaly intrusion detection system using k means and RBF kernel function', *Procedia Computer Science*, Vol. 45, No. 39, pp.428–435.

Sain, H. and Purnami, S.W. (2015) 'Combine sampling support vector machine for imbalanced data classification', *Procedia Computer Science*, Vol. 72, No. 1, pp.59–66.

Sánchez, D. and Trillas, E. (2012) *Measures of Fuzziness under Different Uses of Fuzzy Sets*, in Greco, S. et al. (Eds.) pp.25–34, Springer Berlin Heidelberg, Berlin, Heidelberg.

Sun, Z. et al. (2015) 'A novel ensemble method for classifying imbalanced data', *Pattern Recognition*, Vol. 48, No. 5, pp.1623–1637.

Tahir, M.A. et al. (2012) 'Inverse random under sampling for class imbalance problem and its application to multi-label classification', *Pattern Recognition*, Vol. 45, No. 10, pp.3738–3750.

Tavallaee, M. et al. (2009) 'A detailed analysis of the KDD CUP 99 data set', *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*.

Wang, X-Z. et al. (2015) 'Fuzziness based sample categorization for classifier performance improvement', *Journal of Intelligent and Fuzzy Systems*, Vol. 29, No. 3, pp.1185–1196.

Zadeh, L.A. (1968) 'Probability measures of fuzzy events', *Journal of Mathematical Analysis and Applications*, Vol. 23, No. 2, pp.421–427.

Zhang, J. et al. (2015) 'Detecting anomalies from big network traffic data using an adaptive detection approach', *Information Sciences*, Vol. 318, pp.91–110.

Zhang, Z. et al. (2016) 'Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data', *Knowledge-Based Systems*, Vol. 106, pp.251–263.

Zhu, X. and Goldberg, A.B. (2009) 'Introduction to semi-supervised learning', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 3, No. 1, pp.1–130.