International Journal of e-Collaboration
Volume 19 • Issue 3

Mutual Clustered Redundancy and Composite Learning for Intrusion Detection Systems

Thotakura Veeranna, Jawaharlal Nehru Technological University, India*
R. Kiran Kumar, Krishna University, India

ABSTRACT

In the area of cyber space security, intrusion detection is a challenging task which aims at the provision of security from various malicious attacks. Hence, this paper proposes a two-stage hybrid intrusion detection system (IDS) mechanism to identify between normal and attack activities. The proposed mechanism is an integrated form of two simple and effective machine learning algorithms; they are support vector machine (SVM) and composite extreme learning machine (CELM). The first stage aims to distinguish the normal activities from abnormal activities and employed SVM. Next, the second stage employs CELM for the detection of different types of attacks . Further, aiming over training data, a clustering followed by duplicate connections removal and duplicate features removal is accomplished through fuzzy C-means clustering, correlation, and mutual information respectively. The proposed method applied eventually on the standard benchmark dataset NSL-KDD and the real modern UNSW-NB15 dataset. The performance analysis validates through accuracy, false alarm rate and computational time.

KEYWORDS

Correlation, Extreme Learning Machine, FCM, Intrusion Detection, Mutual Information, NSL-KDD, Polynomial Kernel, RBF kernel, Support Vector Machine, UNSW-NB15

1. INTRODUCTION

In recent days, the advancement in the technology, especially with the popularity of internet, a huge demand has been arisen for different applications related to entertainment, electronic communication etc., which are the part of daily life. However, this massive advancement in the computer networks technology increased the vulnerability of cyber-attacks (Buczak & Guven, 2016). Hence the design of cyber-security has been attracted the researchers from both academic and industry. The first line of defense methods for different applications or organizations is several types. They are namely user authentication, data encryption, malware prevention and firewalls. All these methods can ensure a

DOI: 10.4018/IJeC.316772 *Corresponding Author

secure communication and prevents the organizations and enterprises from the victims of cyber-attacks (Al-Jarrah et al., 2015). To enter into an organization, attackers exploit the vulnerabilities much deliberately over the target system and launch various types of attacks those may lead to several problems like information leakage, system rupture etc. With the progress in time, these attacks threaten the availability, integrity and confidentiality of cyber systems. Hence there is a need of an effective Cyber-attacks detection mechanism to protect the systems from different kinds of security attacks.

Intrusion Detection System (IDS) (Inayat et al., 2016; Hubballi & Suryanarayanan, 2014; Khraisat et al., 2020) is one possible solution that can protect the network actively from illegal and external attacks. The main aim of IDS is to ensure the security for systems and to detect the abnormal events phenomena. Moreover, the IDS can also enhance the security and reliability of systems by analyzing and identifying the behavior of malicious activities those enter into the system. Frankly to say, IDs are the widely employed methods in several distributed systems (Wang et al., 2016), perceiving the intrusions and then taking fast countermeasures to prevent from further spreading and infections. According to the detection mechanism, the IDSs are categorizes into two categories such as the Anomaly and Misuse Detection (Joldzic et al., 2016). In cyber-security, the anomaly is defined as an event that has a deviated behavior from normal behavior. The anomaly based IDSs perform better in the detection of novel attack types, but they could not avoid the larger false positive rate (Villalba et al., 2015). On the other hand, the misuse based IDSs (Hubballi & Suryanarayanan, 2014) can detect the legitimate activities form malicious ones because they works based on the known patterns. In this kind of system, the patterns of legitimate activities are stored and if the new activity is found to have a deviated pattern from legitimate patterns, then it is identified as attack. Though this category is reliable for the detection of know attacks, it is not effective in the detection of unknown attacks.

Unfortunately, due to the presence of sophisticated attackers, new vulnerabilities and threats are emerging rapidly. So, to deal with these sophisticated attacks, there is a need of an effective IDS mechanism which has put forward in recent research. Machine Learning (Du, Wang, Chen et al, 2018; Du, Wang, Xia et al, 2018; Mishra et al., 2018) is one of them which can be applied for both misuse and anomaly detection based IDSs. By the analysis of network traffic that was passing through central network nodes, IDSs has to identify the genuine traffic from malicious traffic. Along with this detection, it also has to infer the particular class of attack incurred in the protected system. However, in most of the traffic connections, only a small portion indicates the malicious behavior while most of the IDS considered an entire traffic for training. This kind of processing leads to the difficult of huge computational burden over the IDS and also shows impact on the detection rate. Along with this problem, there exists one more problem in the past developed IDSs models. Most of them applied only one classifier which may not be strong enough to build good IDS. Hence, recent researchers were concentrated on the ensemble of classifiers for IDSs (Feng et al., 2018; Salo, Nassif, & Essex, 2019). For example, modelling the IDS with a single classifier over different subsets could results in different performances, but, the ensemble method could average the output of ensemble classifier and hence it becomes a better option. The main aim of ensemble IDS is to integrate different classifiers and then make a better decision reading the class of submitted input traffic (Pham et al., 2018).

Based on this inspiration, we propose an ensemble IDS by integrating two classifiers namely SVM and the CELM. Here the SVM is employed for misuse detection and CELM is employed for anomaly detection. For a given input traffic, the developed system extracts the features through Mutual Redundant Feature Selection (MRFS) method and they are processed through SVM algorithm first. If the SVM algorithm classified it as attack, then it was again processed through CELM to find the specific attack class. We employs binary SVM classifier which assigns only two labels, positive and negative. Further, the ELM is extended by including two kernels namely polynomial and Radial Basis Function and hence it is called as CELM. The proposed method is compared with several existing methods after is validation through different datasets like UNSW-NB15 dataset and KDD-Cup99. The performance is measured through Accuracy and False alarm rate measures

The remaining paper is structured as follows; in section 2, we review different methods those were proposed in earlier. Under this review, we approach both single and ensemble IDSs. Then the details of proposed method are explored in section 3. Section 4 illustrates the details of experimental and comparative analysis and the concluding remarks are shown in Section 5.

2. LITERATURE SURVEY

Researchers have concentrated on the design of both anomalies and misuse based IDS through individual classification methods. However, they are susceptible to different attacks and failed to ensure best possible attack detection. Due to this reason, recently the researchers have diverted their concentration on the design of Hybrid approaches (Folino & Sabatino, 2016) which introduces so many new challenges in IDS. ML algorithms are widely employed by researchers for the design of anomaly based IDS (Tsai et al., 2009). The main objective behind the ensemble method is to enhance the performance of IDS by integrating different data mining and ML algorithms (Peddabachigari et al., 2007).

Wang et al. (2010) proposed an ensemble method for the anomaly based Intrusion detection. This method combined two ML algorithms namely Artificial Neural Networks (ANN) and Fuzzy Clustering (FC). FC is adopted for the creation of different training sets and ANN is adopted for the training of created models. Finally they applied fuzzy aggregation module to find the average results all models. Experiments are conducted through the KDD Cup 99 dataset and performance is measured through detection stability and precision.

Kuang et al. (2014) developed an IDS model based Kernel Principal Component Analysis (KPCA), SVM algorithm and Genetic Algorithm (GA). The main intention of KPCA is to lessen the dimensions of input network traffic and SVM is to detect the network activities. At the SVM mode, they employed Radial Basis Function kernel and the parameters are optimized through Genetic Algorithm.

De la Hoz et al. (2014) developed a hybrid IDS model by combining a multi-objective optimization approach called as NSGA-II with classification method. For the purpose of feature selection, they used NSGA-II (Deb et al., 2002) and for classification, they employed Growing Hierarchical Self-Organizing Maps (GHSOMs) (Rauber et al., 2002) which classifies both anomalies and attacks. Experiments are conducted through two datasets such as DARPA and NSL-KDD that contains the features and labeled attacks.

Eesa et al. (2015) presented a hybrid IDS model by combining two algorithms such as Cuttlefish Algorithm (CA) (Eesa et al., 2014) and Decision Tree (DT). CA is applied for feature selection which performs searching operation and asserts an optimal subset of features. Next the DT is applied for classification that judges the characteristic of obtained feature through CA. For experimental validation, they used KDD Cup 99 dataset and the performance is measured through false alarm rate and accuracy.

Hoz et al. (2015) proposed an anomaly based model by hybridizing three algorithms namely Probabilistic Self Organizing Maps (PSOMs), Fisher Discriminant Ratio (FDR) and PCA. In their method, the FDR and PCA are aimed at the discovery of feature selection by suppressing noises, PSOMs are aimed at the modelling the feature space and ensure a perfect discrimination between normal and malicious connections. The detection capabilities are altered without perform repetitive training but only with altering the probable activation units. .

Singh et al. (2015) handled the Intrusion detection through a new model called as Online-Sequential ELM (OS-ELM). OS-ELM used alpha profiling to lessen the time complexity by discarding the redundant features through an ensemble filter, consistency and correlation based feature selection technique. In the training phase, the time complexity is reduced through the redundant feature removal through beta profiling. For experimental validation, they used NSL-KDD dataset and Kyoto dataset.

A. A Aburomman and M. B. I. Reaz (2016) developed an ensemble IDS method based on three methods namely Particle Swarm Optimization (PSO), SVM and K-nearest neighbor (k-NN) (Canbay & Sagiroglu, 2016). In their, they were used totally six k-NN experts and ix SVM experts to train

the system. Further, they generated two ensemble classifiers by mixing the opinions of 12 experts through weighted majority voting scheme. For the generation weight, they employed PSO algorithm. For the optimization of parameters of first ensemble classifier, they employed mutual PSO parameter selection while for second ensemble classifier; they employed Local Unimodal Sampling (LUS). KDD Cup 99 dataset is used for experimental validation

Jamal Hussain et al. (2016) proposed a two-stage hybrid method for IDS using SVM and ANN. In their method, SVM classifier is employed in first stage for anomaly detection and ANN is employed as a second stage classifier for misuse detection. The first stage detects abnormal activities that can be called as attack. The second stage further analyses if there is a known attack and classifies into four attack types. LIBSVM (Chang & Lin, 2011) is employed which is an integrated tool for SVM and can handle binary of multi-class SVM. For experimental validation, they used NSL-KDD and KDD cup 99 datasets.

W. L. Al-Yaseen et al. (2017) designed a real and multi-level hybrid IDS model that analyzes the data and classify network traffic into normal and abnormal classes. They used the combination of SVM and ELM algorithms for modelling the system. A modified k-means algorithm is also employed to construct a high quality dataset that contributes to build a small training dataset. For experimental validation, hey employed KDD Cup 99 dataset.

S. M. H. Bamkan et al. (2016) presented an effective IDS mechanism called as time-varying chaos PSO (TVCPSO) which simultaneously optimizes parameters and select an effective feature set. TVCPSO works bot for multiple criteria linear programming (MCLP) and SVM. The performance is assessed through KDD Cup 99 and NSL-KDD datasets.

L. Li et al. (n.d.) proposed a two-step hybrid IDS framework based on binary classification (C4.5 algorithm was employed) (Khraisat et al., 2018) and K-NN. Step 1 identifies the exact class of input traffic connection through the binary classification followed by an aggregation module. Step concentrates on the class uncertainty to further determine their class through K-NN algorithm. By combining these two steps, the experimental validation is done through NSL-KDD dataset.

Y. Tian et al. (2018) developed a robust and sparse anomaly detection mechanism that introduces Ramp loss Function to the original one-class SVM called as Ramp-OCSVM. This approach is a semi-supervised algorithm that considers the advantage of non-convex property y of ramp loss function. Next, Concave-Convex Procedure (CCP) is employed to solve the non-differentiable non-convex optimization problem. The performance is assessed through UNSW-NB15 and NSL-KDD datasets.

B. A. Tama et al. (2019) proposed a two-stage IDS mechanism based a hybrid feature selection and two level ensemble classifier. The feature selection composed of three methods; they are PSO, GA and Ant Colony Optimization (ACO) and they are used to lessen the size of feature set. Features are chosen based on the classification performance of reduced error pruning tree (REPT) classifier (Gaikwad & Thool, 2015). After the extraction of features, they are processed through two Meta-classifiers, they are bagging and forest. The performance is assessed through UNSW-NB15 and NSL-KDD datasets.

To improve the scalability in IDS, M. A. Khan et al. (2019) proposed a two-stage scalable and hybrid IDS framework with the help of Convolutional LSTM and Spark ML. In their method, the first stage is employed through Spark ML to identify anomalies while the second stage is employed through Conv-LSTM network. Evaluation is done through ISCX-UNB dataset (Shiravi et al., 2012) through 10-fold cross validation.

P. Salo et al. (2019) proposed a hybrid dimensionality reduction method by combining PCA and Information Gain (IG). Further, an ensemble classifier is designed by combining SVM and Instance based learning algorithm and Multilayer perceptron (MLP). Evaluation is done through ISCX 2012, NSL-KDD and Kyoto 2006+ datasets.

Wathiq et al. (2017) proposed a method called real-time multi-agent system for an adaptive IDS (RTMAS-AIDS) to allow the IDS to adapt to unknown attacks in real-time, and this method applied a hybrid SVM and ELM to classify normal behavior and known attacks. An effective SVM-based

ID algorithm was presented by Tao et al. (2018) to identify intrusions, which obtained great results. In addition, the improved algorithms of ELM and SVM were widely used for ID applications. These advances have achieved a great performance for detecting and reporting malicious attacks. Nevertheless, the better accuracy and efficiency of the prediction model is still the first purpose of an IDS.

Even though an extensive research is incurred even in the hybrid designs, there are several limitations. First, there is no exact information provision. Some methods report only the occurrence of attacks but not their type. Actually there is a need of exact attack information for network administrators such that they can take relevant actions. In some methods, it is observed that the Lower detection performance, especially at minor attacks like U2R and R2L. Actually, they have serious effect on the network than the major attacks like DoS, and Probe. Hence, the identification of such kind of attacks is very important and most of researchers not focused in that direction. The last limitation is a huge number of parameters at hybrid designs. The ensemble methods have many parameters for which setting an optimal value is not that much easy. Even though some methods applied metaheuristic algorithms for the optimization purpose, they increases the training time and the values obtained are may or may not optimal. Unoptimized values show a serious effect in the performance by increasing false positives. Hence the parameters reduction is necessary.

3. PROPOSED APPROACH

3.1 Overview

In this section, we describe the complete details of developed hybrid IDS mechanism. The proposed IDS is designed as an integrated model by combining two meat classifiers namely SVM and CELM. Here, we employed a binary version of SVM classifier which classifies the input data into only two classes. Hence, the SVM is employed at first phase for anomaly detection, i.e., defection of abnormal behaviors from normal behaviors. In the second stage, we employed the modified version of ELM called as CELM for attacks classification, i.e., misuse detection. If the SVM is classified the input traffic connection as attack, then it was fed to second stage CELM for further class identification. The CELM is formulated as a weighted combination of two different kernels such as polynomial kernel and RBF kernel. Along with these aspects, we also model new methods at different stages. In the first phase, the raw network traffic is subjected to pre-processing. Next, we apply FCM clustering at training phase to cluster the entre training data. For connections selection in each cluster, a correlation process is employed. Finally, to represent each class with an effective set of features, we applied MRFS. Even though there exist minimal redundancy for feature selection, it subjects to more features elimination those results in poor recognition performance, especially for new attacks. Figure 1 shows the schematic of proposed IDS mechanism.

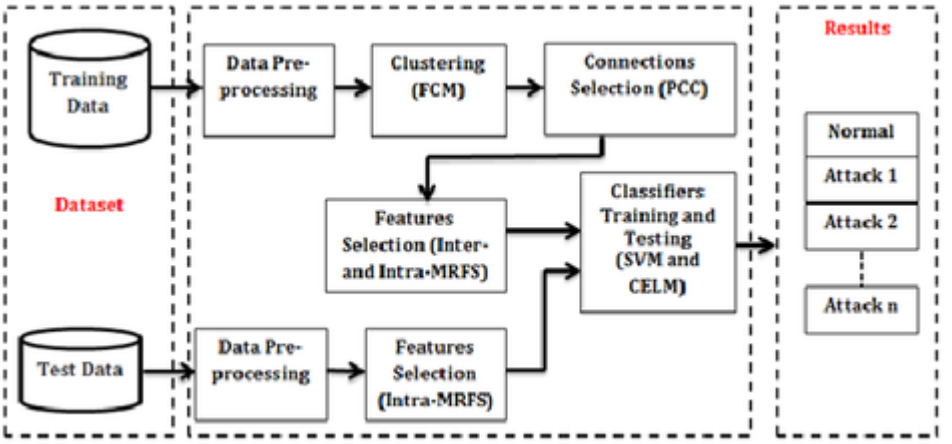
3.2 Pre-Processing

The main aim of data pre-processing is to remove inconsistency, noise and incompleteness of network traffic. Since the raw network traffic is of so clumsy, preprocessing is required which makes the data clear and informative. Moreover, the raw network traffic is not present in uniform format. For example, the features of NSL-KDD are all not in numerical format, some are in numerical form and some are in symbolic form. Hence it needs to be transformed into a suitable format that includes the removal of duplicate features and replacement of incomplete or missing features with zeros. The generalized algorithm for data pre-processing is shown below;

Step 1: Consider the dataset X with size $M \times N$, where M is total number of connections and N is total number of features used to represent each connection.

Separate the corresponding Colum or Row which is not in uniform format

Figure 1. Proposed hybrid IDS model



$$C_i = X\left(:,i\right) \text{ or } C_i = X\left(i,:\right) \tag{1}$$

Defines the total number of features those are in uneven format. Let they are $Feature\_X$. Next find out the total number of occurrences of each and every sub-feature in the C_i by comparing with defined features as

$$M_i = \sum_{i=1}^{L} strcmp\left(Feature\left(i\right), Feature\_X\right) \tag{2}$$

Measure the probability of each feature as

$$PF_i = \frac{M_i}{Length\left(Feature\_X\right)} \tag{3}$$

Where M_i is the total number of occurrences of feature *i* and $Length\left(Feature\_X\right)$ denotes the total size of respective column.

Step 5: Replace the probability values of i^{th} feature in their respective positions in the dataset X.

For other datasets, if we observe the incomplete connections, then the connection is completed by adding zeros in sufficient number. Similarly for the datasets which have connections with abnormal values like NaN and Infinity, they are replaced with 0's.

3.3 Clustering

The main aim of clustering is lessen the computational complexity and computational time at the testing phase. If the system is trained without clustering, for every testing phase, the entire database need to be scanned to find the best match with test input traffic connection. Due to this process, the system takes more time for testing if the size of test data is large. Clustering is one possible solution in which the training data will get clustered into several clusters. Due to this clustering, at the time

of testing, we can avoid the problem of entire database scanning to find best match. At here, the total number of clusters into which the data needs to get clustered is completely user dependent. For instance, to cluster the NSL-KDD dataset there is a need of five clusters and for Kyoto 2006+, there is a need of two clusters. For clustering purpose, we employed the most popular FCM algorithm that was initially proposed by Bezdek (1984). For FCM, the inputs are entropy features which are computed for every traffic connection of training data. For example, if the dataset size is size $M \times N$, where M is total number of connections and N is total number of features used to represent each connection, each connection is represented with single entropy value. FCM clusters the entire dataset into C clusters based on these M entropies. According to FCM, the objective function is modeled as

$$J_m = \sum_{i=1}^{M}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \tag{4}$$

Where *m* is a real number greater than 1, u_{ij} is the membership function between i^{th} entropy value and j^{th} cluster center, M is the total number of connections and C is total number of clusters. . denotes the normalization function which defines the similarity check between any measured entropy values and cluster center. The cluster center and membership function are iteratively updated and mathematically they are calculated as

$$u_{ij} = \frac{1}{\sum_{c=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_c \right\|} \right)^{\left(\frac{2}{m-1} \right)}} \;\&\; c_j = \frac{\sum_{i=1}^{M} u_{ij}^{m} . x_i}{\sum_{i=1}^{M} u_{ij}^{m}} \tag{5}$$

The iteration process is terminated when the following condition met

$$\max_{ij} \left\{ \left| u_{ij}^{(q+1)} - u_{ij}^{q} \right| \right\} < \varepsilon \tag{6}$$

Where ε is the termination threshold which lies in between 0 and 1 and *q* indicates the iteration number. Based on this process, we can state that the clustering of connections with similar semantics will reduce the unnecessary computational burden over the system. Moreover, we also analyzed that the different traffic connections of same class have only small deviations in their feature values. Clustering of such kind of connections will improve the efficiency and lessens the computational burden over the system. After the completion of clustering, we focus on the removal of duplicate connections in every cluster and it was accomplished through the correlation of traffic connections.

3.4 Correlation Based Connections Selection

The main aim of correlation based connections selection is to remove redundant connections in training data. After the completion of clustering, each cluster has traffic connections with similar characteristics. Hence we accomplish correlation based duplicate connections removal. For this purpose, we applied Person Correlation Coefficient (PCC) which determines the linear relations between two variables (Nguyen et al., 2010). Consider two connections, X and Y from any cluster, the PCC is calculated as

$$r\left(X,Y\right)=\frac{cov\left(X,Y\right)}{\sigma_X\sigma_Y} \tag{7}$$

Where

$$cov\left(X,Y\right)=\sum_{i=1}^{N}\left(x_i-\bar{X}\right)\left(y_i-\bar{Y}\right) \tag{8}$$

Where σ_X and σ_Y are the standard deviations of connections X and Y respectively. And $\bar{X}$ and $\bar{Y}$ are the mean of two connections X and Y respectively. The value of PCC fall into the closed interval of -1 and 1, where the both values denotes that the two connections are strongly correlated while the mid values, i.e., 0 indicates that the two connections are weakly correlated. Based on the value of PCC, we decide whether the connection is duplicate or not. For this purpose, we fix a threshold value and if the PCC between two connections (let X and Y) is greater than the threshold, then we consider the second connection (i.e., Y) us duplicate and we remove it from the cluster. The threshold computation is done by averaging the PCC values of every connection with respect every connection in the cluster, as

$$T=\frac{1}{length\left(Q\right)}\sum_{i=1}^{P}\sum_{j=1}^{P}r_{ij} \tag{9}$$

Where r_{ij} is the PCC between two connections *i* and *j* and Q is the size of cluster. After the computation of threshold, the PCC of every two connections is compared with the threshold T to find out the duplicate connections. If the value of r_{ij} is found to be greater than the threshold, then the j^{th} connections is determined as duplicate connection and it was kept in the removal list. After the completion, the connections those have maximum appearance in the removal lists are only removed from the cluster. In this manner, we eliminate the duplicate connections from every cluster.

3.5 Feature Selection Through Mutual Redundancy

The main aim of feature selection is to determine the effective set of features those contribute more information of a class or cluster. For this purpose, we employed a Mutual Redundancy based feature selection (MRFS) in two ways, they are Inter and Intra. The Intra_MRFS is applied within the connection and the Inter_MRFs is applied between the connections. For Intra_MRFS computation, initially each connection is divided into several blocks through sliding window. For feature selection process, we followed our earlier contribution (Veeranna & Reddy, n.d.) in which we apply Sliding window as a preprocessing and Duplicate Mutual Information (DMI) for feature selection. Consider $B_1^U,B_2^U,\ldots,B_Z^U$ be the Z number of blocks obtained after partitioning the connection U through sliding window. After the sliding windowing, we compute the MI (Roulston, 1999) between as well as within connections. Consider two connections U and V and the respective blocks are $B_1^U,B_2^U,\ldots,B_Z^U$ and $B_1^V,B_2^V,\ldots,B_Z^V$, then the Inter_MI is computed as follows;

$$I\left(B_i^U,B_i^V\right)=\sum_{f_k\in B_i^U}\sum_{f_l\in B_i^V}p\left(f_k,f_l\right)\log\left(\frac{p\left(f_k,f_l\right)}{p\left(f_k\right)p\left(f_l\right)}\right) \tag{10}$$

Where B_i^U and B_i^V are the blocks of the data connection U and data connection V respectively and f_k and f_l are the features of i_{th} block in the connection U and i_{th} block in the different connection V respectively. Next, Intra_MI is computed as follows

$$I\left(B_i^U, B_j^U\right) = \sum_{f_k \in B_i^U} \sum_{f_l \in B_j^U} p\left(f_k, f_l\right) \log\left(\frac{p\left(f_k, f_l\right)}{p\left(f_k\right) p\left(f_l\right)}\right) \tag{11}$$

Where B_i^U and B_j^U are the two blocks of the data connection U and f_k and f_l are the features of i_{th} block and j_{th} block respectively. In the above expression, we compute the Mutual Dependency between the blocks of same connection and based on obtained MI values, we decide which blocks are mutually dependent and mutually independent. The two blocks are selected which has stronger mutual dependency within the connection and the features of those blocks are considered as required subset of features. However, this process eliminates some features those significance. To regain such kind of features, we measure duplication between eliminated features and selected subset of features through the following DMI

$$DMI = \frac{I\left(f_i; f_s\right)}{I\left(C; f_i\right)} \tag{12}$$

Where $I\left(f_i; f_s\right)$ is the Mutual information between the feature f_i left in the connection and the feature f_s selected in the subset. Next, $I\left(C; f_i\right)$ explores the MI between class and feature f_i. With the help of Eq.(16), we can state that the features are selected those have more information to contribute with respect to both class and neighbor features. The final set of features is obtained based on the following expression (Ji et al., 2016);

$$I_{MI} = \arg\max_{f_i \in U}\left(I\left(C; f_i\right) - \frac{1}{\left|B\right|}\sum_{f_s \in B} DMI\right) \tag{13}$$

In the case of $I\left(C; f_i\right) = 0$, the corresponding feature f_i is eliminated permanently. On the other hand, if the features f_i and f_s are highly related, then the feature f_i contributes to redundancy. For this purpose, we have kept a threshold φ and the obtained I_{MI} is compared as follows;

1. If $I_{MI} < \varphi$, then the feature f_i constitutes to redundancy with respect to class C because it may consequences to less MI between the selected features f_s and class C.
2. If $I_{MI} = \varphi$, then the feature f_i constitutes to redundancy with respect to class C because it don't carry any additional information about the class C.
3. If $I_{MI} > \varphi$, then the feature f_i constitutes to relevancy and it have more contribution to class C, because it can provide some additional information about class C and hence it is added to the selected features subset.

3.6 Ensemble Classifier

The proposed ensemble classifier is a combination of two classifiers such as SVM and CELM. The SVM is applied as a first stage classifier while CELM is applied as a second stage classifier. At the training phase, the entire features are grouped into two sets, they are normal feature set and attack feature set. In the case of NSL-KDD, the attack set consists of features belongs to four clusters, they are DOS, Probe, R2L and U2R. These two feature sets are processed through SVM algorithm such that it can assign one label for normal set and another label for attack set. Next, to train the CELM model, we use only attack features and segregation of them into sub-classes. At the testing phase, for a given input traffic connection, initially, the features are extracted through Intra_MRFS and then they are fed to SVM algorithm to determine whether it belongs to normal or attack class. If SVM is classified it as attack, then they are fed to CELM to determine the further sub-class, i.e., the specific attack class. The details of SVM and CELM are explored in the following subsections.

3.6.1 SVM

The SVM is initially introduced by (Boser et al., 1992). Basically, the SVM increase the samples size such that it can separate them effectively. Hence, instead of general trend towards the dimensionality reduction, SVM follows an opposite process and increase the size of features. The main idea is to determine a hyperplane to put the samples from class inside it. SVM employs kernel functions that postulate the linear and non-linear features and hence it is able to construct a separating plane that is implicitly defined by the kernel function. Here, we employed LIBSVM for classification purpose at first stage. For a given input feature vector $x_i \in R^n, i = 1,2,\ldots l$, which belongs to two classes, SVM produces an output vector $y_i \in R^l$ such that $y_i \in \left\{-1,1\right\}$. According to the SVM, the decision function is modeled as

$$f\left(t\right) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i K\left(x_i, x\right) + b\right) \tag{14}$$

Where

α_i = the coefficients of Lagrange multiplier of the *i*th feature,
$K\left(x_i, x\right)$ = kernel function and
b = subjective constant.

The kernel function is formulated as follows;

$$K\left(\left(x_i, x\right)\right) = exp\left(\frac{-\left(x_i - x\right)^2}{\sigma^2}\right), \sigma \in R \tag{15}$$

According to the theory of functionality, as long as the kernel function $K\left(x_i, x\right)$ won't satisfy the condition of mercer's, it cannot be called as positive definite kernel. The mercer's condition is formulated as

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{l}\xi_i \\ &subjected\,to \qquad y_i\left(w^T\phi\left(x_i\right)+b\right) \geq 1-\xi_i \end{aligned} \tag{16}$$

Where $\phi\left(x_i\right)$ maps x_i into higher dimensional space and C>0 is the regularization parameter. Next, w ad b are iteratively tuned to optimize the problem.

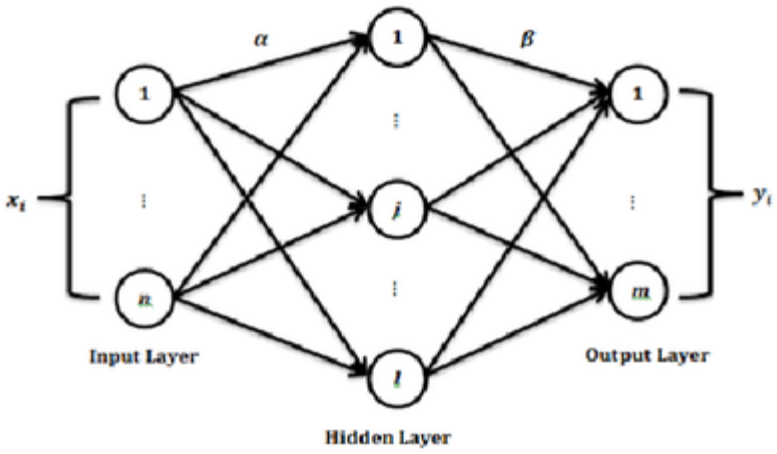
3.6.2 CELM

An ELM is one of the most effective machine learning algorithms that was initially developed by Huang et al. (Huang et al., 2004; Huang et al., 2006). ELM is a feed-forward neural network based algorithm which has only one hidden layer. In the traditional machine learning algorithms, there is a need of larger number of parameters setting thus it can yield a local optimal solution. But, in ELM, there is no need of such kind complex and larger number parameters settings, as it needs only to set the number of hidden nodes in the network. Moreover, ELM also don't seek the adjustment of weights of input layer and the bias of hidden layer, thus it is easy to get a global optimal solution (Wu et al., 2017). Hence, the ELM is observed to have faster convergence rate and is much effective in terms of learning performance. The basic schematic of ELM is shown in Figure 2.

Consider the training dataset is represented as $T_r = \left\{\left(x_i, \boldsymbol{t_i}\right), i = 1,2,\ldots,N\right\}$, where $\boldsymbol{x_i} = \left[x_i^1, x_i^2,\ldots,x_i^n\right]$ is the input feature vector and $\boldsymbol{t_i} = \left[t_i^1, t_i^2,\ldots,t_i^n\right]$ is the respective target vector, the main aim is to determine the optimal model for testing job. As shown in Figure 2, $\boldsymbol{y_i} = \left[y_i^1, y_i^2,\ldots,y_i^n\right]$ is the output vector which has to be determined through the ELM model, then it is represented as

$$\boldsymbol{y_i} = \sum_{j=1}^{l}\beta_j g_j\left(\boldsymbol{x_i}\right) = \sum_{j=1}^{l}\beta_j g\left(\pm_j.\boldsymbol{x_i} + \boldsymbol{c_j}\right), i = 1,\ldots,N \tag{17}$$

Figure 2. ELM model Network Structure



Where $\pm_j$ is defined as a weighted vector that shows the weightage between input and hidden layers, β_j is defined as the weighted vector that shows the weightage between hidden and output layers, $g\left(\pm_j.\boldsymbol{x_i}+\boldsymbol{c_j}\right)$ is the activation function of hidden layer and c_j is the bias function. Since the parameters α_j and c_j of the hidden layer follows a random assigning process, and then only there is a need to determine the hidden layer nodes in the ELM model. For the determination of hidden layer nodes and the attached weights β_j, the error between target vector and output vector is approximated to zero, according to the following equation,

$$\sum_{i=1}^{N}\boldsymbol{t_j}-\boldsymbol{y_j}=0 \tag{18}$$

Substitute Eq.(17) in Eq.(18), then

$$\boldsymbol{t_j}=\sum_{j=1}^{l}\beta_j g\left(\pm_j.\boldsymbol{x_i}+c_j\right),i=1,\ldots,N \tag{19}$$

Expand the above equation and the resultant expanded matrix is observed as follows;

$$\begin{bmatrix} g\left(\pm_1.\boldsymbol{x}_1+c_1\right) & \cdots & g\left(\pm_{\boldsymbol{l}}.\boldsymbol{x}_1+c_l\right) \\ \vdots & \ddots & \vdots \\ g\left(\pm_1.\boldsymbol{x}_{\boldsymbol{N}}+c_1\right) & \cdots & g\left(\pm_{\boldsymbol{l}}.\boldsymbol{x}_{\boldsymbol{N}}+c_l\right) \end{bmatrix}_{N\times l}.\begin{bmatrix}\beta_1^T\\ \vdots\\ \beta_l^T\end{bmatrix}_{l\times m}=\begin{bmatrix}t_1^T\\ \vdots\\ t_l^T\end{bmatrix}_{N\times \boldsymbol{m}} \tag{20}$$

Eq.(20) can be written as

$$\boldsymbol{H}\beta=\boldsymbol{T} \text{ and } \beta=\boldsymbol{H}^{+}\boldsymbol{T} \tag{21}$$

Where **H** is the output matrix of hidden layer, β is the weight of hidden layer and T is the output matrix of target vector. $\boldsymbol{H}^{+}$ is the Moore-Penrose generalized inverse of H matrix, is derived as follows;

$$\boldsymbol{H}^{+}=\boldsymbol{H}^{\boldsymbol{T}}\left(\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\right)^{-1} \tag{22}$$

However, the ELM has relatively low performance when it was applied to some unknown datasets. Hence, an improved version of ELM is introduced by inserting a kernel parameter and it is named as Kernalized ELM (KELM). The output expression of KELM is expressed as

$$f\left(\boldsymbol{x}\right)=h\left(\boldsymbol{x}\right)\beta=h\left(\boldsymbol{x}\right)\left(\boldsymbol{H}^{\boldsymbol{T}}\left(\frac{\boldsymbol{I}}{\boldsymbol{C}}+\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\right)^{-1}\right) \tag{23}$$

Where C is the penalty parameter and I is an identity matrix. The Kernel function of KELM is defined as

$$KELM_{i,j} = h\left(\boldsymbol{x_i}\right)h\left(\boldsymbol{x_j}\right) = K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) \tag{24}$$

Then the Eq.(23) is modified as

$$f\left(\boldsymbol{x}\right) = \begin{bmatrix} K\left(\boldsymbol{x}, \boldsymbol{x}_1\right) \\ \vdots \\ K\left(\boldsymbol{x}, \boldsymbol{x}_N\right) \end{bmatrix}.\left(\boldsymbol{H^T}\left(\frac{\boldsymbol{I}}{\boldsymbol{C}} + \boldsymbol{HH^T}\right)^{-1}\right) \tag{25}$$

In KELM, the kernel function has great influence on the performance. Moreover, the determination of an appropriate kernel function is also required. In this work, we formulated a Composite KELM by combining two kernels; they are RBF kernel and polynomial kernel.

The polynomial kernel is a generalized kernel function which can was used in so many algorithms due to its generalization capability and weal learning ability (Smits & Jordaan, 2002; Tian, Li, Wang et al, 2018). Polynomial kernel function is a global kernel function and it is expressed as

$$K_p\left(\boldsymbol{x}, \boldsymbol{x_j}\right) = \left(\boldsymbol{x}.\boldsymbol{x_i} + b\right)^p \tag{26}$$

Figure 2 shows the curve of polynomial kernel with different values of b and the p value is set as 2. From this figure (3), we can see that the polynomial kernel function increases with an increase in the input. Moreover, the sample points both from near and far away from the test point have an influence on the output of polynomial kernel function. This relation shows strong generalization capability of the polynomial kernel function. But, the test point has no apparent learning capability that reveals the weak learning capability of polynomial kernel function.

Unlike the polynomial kernel, the RBF kernel function has weak generalization ability and strong learning ability. It is a typical local kernel function. RBF kernel function is a local kernel function and it is expressed as

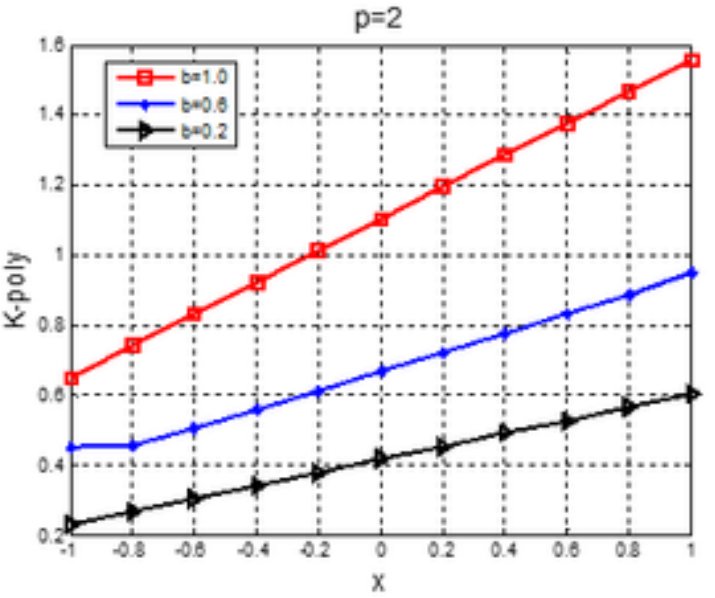
$$K_R\left(\boldsymbol{x}, \boldsymbol{x_j}\right) = exp\left(-\frac{\boldsymbol{x} - \boldsymbol{x_i}^2}{2\sigma^2}\right) \tag{27}$$

Based on these two kernels, we modeled a new kernel function called as Composite kernel which is a weighted combination of polynomial and RBF kernels. The equation of newly modeled kernel function is expressed as

$$K_C\left(\boldsymbol{x}, \boldsymbol{x_j}\right) = w_1.K_p\left(\boldsymbol{x}, \boldsymbol{x_j}\right) + \left(1 - w_1\right).K_R\left(\boldsymbol{x}, \boldsymbol{x_j}\right), w_1 \in \left[0,1\right] \tag{28}$$

Since the polynomial kernel has weak learning ability, we included RBF kernel to overcome that problem. On the other hand, the drawback of RBF kernel, i.e., weak generalization capability is solved through strong generalization capability of polynomial kernels. Hence our hybrid kernel

Figure 3. polynomial kernel curves with different values of b



based ELM can provide more generalization capability along with strong learning ability. This kind of machine learning is robust and can show its effectiveness even in the unknown datasets.

4. EXPERIMENTAL ANALYSIS

To show the effectiveness of proposed hybrid IDS Model, we conduct a vast set of experiments over different datasets and the performance is analyze at every dataset. Initially, we explain the details of datasets. Next we explore the details of observed results and performance metrics. Finally we alleviate the effectiveness of proposed model by comparing the results obtained through existing methods.

4.1 Datasets and Settings

For simulation, totally we considered two datasets; they are NSL-KDD and UNSW-NB15 dataset. Initially the details of UNSW-NB15 are explained and then the details of NSL-KDD.

4.1.1 UNSW-NB15

The clean UNSW-NB15 dataset (Moustafa & Slay, 2015) is created by IXIA Perfect Strom tool in the cyber range lab of the Australian Center for Cyber Security (ACCS). The main aim of this dataset generation is create a hybrid combination of real modern normal activities and synthetic contemporary attack activities. The total number of attack present in this dataset are nine, they are namely Worms, Shellcode, Reconnaissance, Generic, Exploits, DoS, Backdoors, Analysis, Fuzzers. The total number of features used to represent each traffic connection is 49 along with one class label. However, most of the researchers used only 42features. Among these 42 features, 39 are numeric and three are nominal features. The numerical features are of belongs to three types, they are binary, float and integer. The three features those are not numerical are *state (f4), service (f3),* and *protocol (f2)*. In this dataset, there exists two subsets, they are training and testing. The total number of connections present in Training set are 1,75,341 and the total number of connections present in test set are 83,332. The details of number of connections present in these sets are demonstrated in Table.1.

Table 1. UNSW-NB15 dataset statistics

Class/Set		Training	70% of Training	Testing	70% of Testing
Normal		56,000	39,200	37,000	25,900
Attacks	**Generic (A1)**	40,000	28,000	18,871	13,210
	Exploits (A2)	33,393	23,375	11,132	7792
	Fuzzers (A3)	18,184	12,729	6062	4243
	DoS (A4)	12,264	8585	4089	2862
	Reconnaissance (A5)	10,491	7344	3496	2447
	Analysis (A6)	2000	1400	677	474
	Backdoor (A7)	1746	1222	583	408
	Shellcode (A8)	1133	793	378	265
	Worms (A9)	130	91	44	31
	Total	1,19,341	83539	45,332	31,732
Total		1,75,341	1,22,739	82,332	57,632

4.1.2 NSL-KDD

The NSL-KDD is a revised version of KDD cup 99 dataset that has been proposed by Tavallaee et al. (2009). This dataset is reconstructed by addressing several problems of KDD cup99 like huge number of redundant records. To group the connections into five groups, the initial dataset was subjected to different classifiers and everyone is labeled with the number of successful estimations. This dataset consist totally five types of classes. They are Normal, DoS, U2R, R2L and Probe. Among these classes, the first one is non-attack and the remaining four are attacks. Each connection of NSL-KDD dataset consists of 41 features. Further, the dataset consist of three different sets, they are KDDTrain^{+}, KDDTest^{+} and KDDTest^{-21}. The initial set, i.e., the KDDTrain+ consist of totally 125973 connections among which the 67343 are normal traffic connections and 58630 are attack traffic connections. In the second set, i.e., KDDTest^{+}, the total number of traffic connections are 22544 among which 9711 are normal traffic connections and 12833 are attack traffic connections. Finally in the KDDtest^{-21} set, the total number of connections present are 11850 out of which 2152 are normal traffic connections and 9698 are attack traffic connections. We conduct cross validation over the KDDTrain^{+} set and also consider the validation test using KDDTest+ and KDDTrain-21 sets. The details of number of connections present in these sets are demonstrated in Table 1.

Table 2. NSL-KDD dataset statistics

Class/Set		KDDTrain^{+}	KDDTest^{+}	KDDTest^{-21}
Normal		67343	9711	2152
Attacks	**DoS**	45927	7458	4342
	U2R	52	200	200
	R2L	995	2754	2754
	Probe	11656	2421	2402
	Total	58630	12833	9698
Total		125973	22544	11850

4.2 Results

Initially, we explain the results of UNSW-NB15 and then we explain the details of NSL-KDD. For both datasets, initially we train the system with the number of specified connections in the above tables. Once the training is complete, we start testing through eh testing connections. After eh completion of testing, a confusion matrix is formulated based on detected results. From that confusion matrix, we measure the performance through several performance metrics. Here we used the performance metrics such as Detection Rate, Precision, F-Score, False Negative Rate (FNR), False Positive Rate (FPR), False Alarm Rate (FAR) and Accuracy.

Table 3 shows the confusion matrix of the results obtained after the simulation of proposed model over UNSW-NB15 dataset. For the simulation purpose, we have considered only 70% of training data and testing data. Actually, the original NSW-NB15 has totally 1,75,341 traffic connections is training set and 82,332 traffic connections in testing set. Here, we aimed to conduct a fivefold cross validation mechanism and hence we have considered only 70% of data both in training and testing phases. At every validation, we remove some traffic connections from the past trained and test sets and adds new

Table 3. Confusion matrix of UNSW-NB15 dataset

Actual/ Predicted	Normal	A1	A2	A3	A4	A5	A6	A7	A8	A9	Total
Normal	**25274**	6	175	261	5	44	14	5	113	3	**25,900**
A1	177	**12098**	506	169	120	56	4	0	80	0	**13,210**
A2	413	17	**6706**	200	155	134	10	14	138	5	**7792**
A3	934	8	80	**3028**	31	64	6	3	88	1	**4243**
A4	224	52	731	261	**1200**	181	31	13	167	2	**2862**
A5	103	6	173	38	37	**1987**	6	0	97	0	**2447**
A6	64	0	82	73	83	37	**119**	0	15	1	**474**
A7	40	0	63	55	74	45	20	**91**	20	0	**408**
A8	60	0	3	9	0	24	0	0	**169**	0	**265**
A9	1	0	13	0	0	0	0	0	0	**17**	**31**
Total	**27920**	**12187**	**8532**	**4094**	**1705**	**2572**	**210**	**126**	**887**	**29**	**57,632**

Table 4. performance metrics of proposed method for UNSW-NB15 dataset

Class/Metric	DR (%)	PPV (%)	FNR (%)	FPR (%)	FAR (%)	F-Score (%)
Normal	97.5820	90.5212	2.4180	9.4788	5.9484	93.9191
Generic	91.5824	99.2734	8.4176	0.7266	4.5721	95.2729
Exploits	86.0615	78.6056	13.9385	21.3944	17.6665	82.1648
Fuzzers	71.3632	73.9667	28.6368	26.0333	27.3350	72.6416
DoS	41.9347	70.3889	58.0653	29.6111	43.8382	52.5577
Reconnaissance	81.2085	77.2690	18.7915	22.7310	20.7612	79.1898
Analysis	25.1169	56.6701	74.8831	43.3299	59.1065	34.8069
Backdoor	22.3020	72.2212	77.6980	27.7788	52.7384	34.0800
Shellcode	63.7755	19.0523	36.2245	80.9477	58.5861	29.3397
Worms	54.8436	58.6234	45.1564	41.3766	43.2665	56.6705

connections those were not used in earlier validations. In this way, we conduct totally five-fold cross validation and the best results are shown in Table 3. Based on these values, the performance is measured through several performance metrics and they are shown in Table 4. From these values, we can observe that the maximum Recall is observed for Normal. The main reason behind this superior performance is twofold; (1) binary classification with SVM classifier. Sine there exists a perfect discrimination between normal and attack traffic, the SVM can classify them effectively. (2) Moreover, if we see in a generalized manner, the entropy of normal traffic connections is too much deviated from the entropy values of attack traffic. Due to this reason, the proposed approach has gained more detection rate for normal traffic. In the case of different attacks, the maximum performance is observed at Generic attack, as its recall rate is 91.5824% which is larger than the remaining recall rates. The minimum recall is observed for the attack called as Backdoor, as it is of approximately 22.3020%. Due to the provision of more number of Generic traffic connections at training, the proposed CELM can identify them effectively. The CELM is an integrated form of two kernels and hence they can find linear and non-linear combinations between training and test set. For a single classifier, the learning ability is weak, if the similar data is trained. Hence the proposed approach is gained a superior performance because; it can provide greater generalization capability and strong learning ability.

A further simulation is carried out by with different kernels of ELM. As the suggested model involves the integration of both polynomial and RBF kernels, we have simulated it with individual kernels, to check the performance. Figure 4 shows the performance comparison of different kernels over UNSW-NB15 dataset. From the mean F-scores we can see that proposed hybrid kernel strategy has gained a lager F-score when compared with individual kernels. The mean F-score of composite kernel is observed as 58.2332% while for polynomial and RBF kernels it is observed as 47.0012% and 51.5641% respectively.

Table 5 shows the confusion matrix of the results obtained after the simulation of NSL-KDD dataset. To construct this matrix, we have simulated the traffic connections of 75% of KDDTrain+ and KDDTest+. This dataset is also subjected to five-fold cross validation by exchanging the traffic connections of each and every class. At every validation, 25% of connections are replaced with new traffic connections in both training and testing sets. The selection of traffic connections is done randomly and there is no specific criterion for this process. Based on the values shown in Table 5, the performance metrics are calculated and demonstrated in Table 6. Based on the performance metrics, we can observe

Figure 4. F-scores of Polynomial, RBF and composite kernels over UNSW-NB15

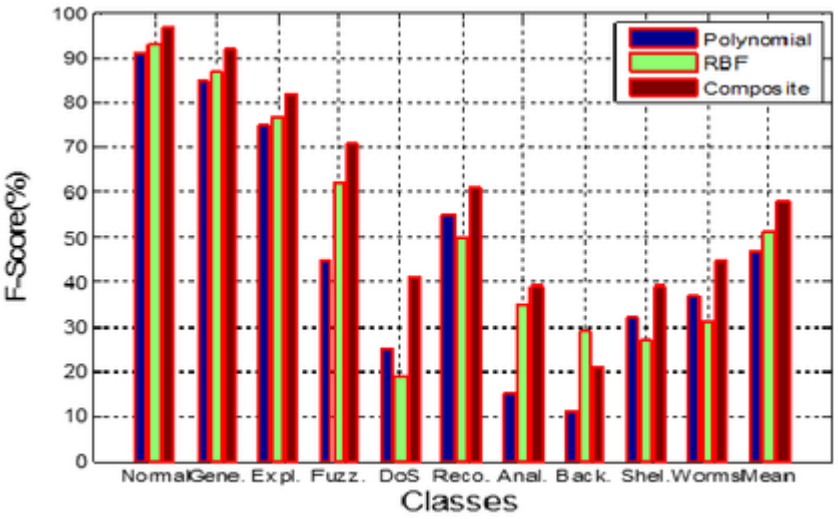


Table 5. Confusion matrix of results from the simulation of KDDTest+ of NSL-KDD dataset

	Normal	DoS	U2R	R2L	probe	Total
Normal	**7075**	111	15	32	50	7283
Dos	59	**5344**	21	22	147	5593
U2R	13	0	**125**	0	12	150
R2L	59	57	9	**1803**	137	2065
Probe	51	55	6	11	**1692**	1815
Total	7257	5567	176	1868	2038	**16906**

Table 6. performance metrics of proposed method for KDDTest+ of NSL-KDD dataset

Class/Metric	DR (%)	PPV (%)	FNR (%)	FPR (%)	FAR (%)	F-Score (%)
Normal	97.1400	97.4923	2.8600	2.5077	2.6838	97.3158
DoS	95.5524	95.9986	4.4476	4.0014	4.2245	95.7750
U2R	83.3385	71.0245	16.6615	28.9755	22.8185	76.6903
R2L	87.3154	96.5247	12.6846	3.4753	8.0800	91.6894
Probe	93.2245	83.0258	6.7755	16.9742	11.8748	87.8301

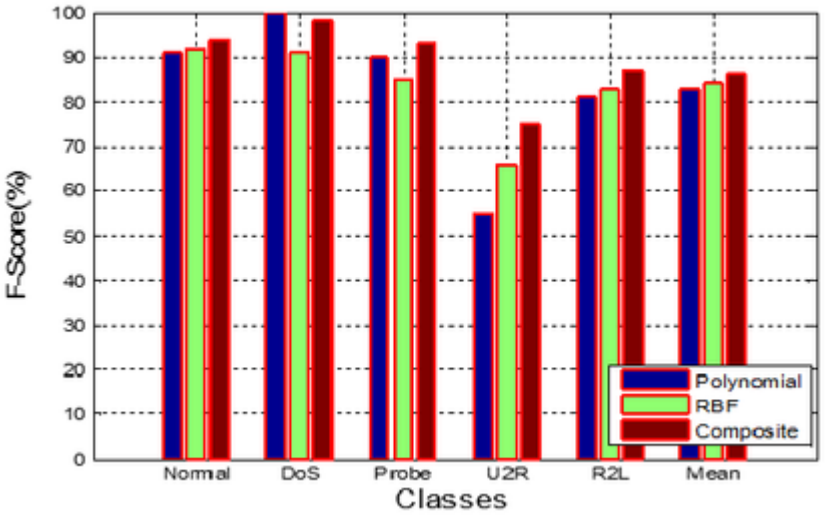
that the maximum recall and precision are observed at the classification of normal class. Further, among the attack classes, the major attacks such as DoS and probe has gained almost equal performance while the minor attacks has gained slightly lower performance. Even though it is slightly lower than the major attacks, they have gained a significant improvement. The major reason behind this improvement is the provision of perfect discrimination between different attacks through strong learning ability.

Figure 5 shows the comparison between different kernels through F-score and mean F-scores. To do this simulation, the ELM is employed with three set of kernels. Initially, it was employed with only polynomial and then only with RBF kernel. Next, the ELM is accomplished with both kernels. At every simulation, the detected results are recorded and performance is analyzed through F-score. From the obtained values, it is clear that the proposed composite mechanism is more effective than the individual kernels because it provides the system strong learning ability and more generalization capability. The mean F-score of composite kernel is observed as 86.4512% while for polynomial and RBF kernels it is observed as 84.2231% and 83.1247% respectively.

4.3 Comparison

Table 7 shows the comparison between of several IDS methods through average accuracy and average FAR. From the comparison, we can see that the maximum accuracy is achieved by GHSOM + NSGA-II (De la Hoz et al., 2014) and it is approximately 99.12%. However, it has been reported that the Average FAR is approximately 2.24%. Though it is reported a maximum detection performance, the accomplishment of NSGA for parameters optimization results in an excessive computational time both at training and testing phases. Next, the recent method proposed in (Tama et al., 2019) employed totally three optimization algorithms for feature selection. This kind of feature selection results in heavy computational burden over the system, because, they are iterative methods and consume more time to get an optimal feature set. Moreover, they can't determine a perfect and discriminative feature set which can improve the detection performance. Hence it has achieved only 91.27% accuracy on the cost of heavy FAR. On the other hand, the method proposed in (Li et al., n.d.), (Hussain et al.,

Figure 5. F-scores of Polynomial, RBF and composite kernels over NSL-KDD



2016) and (Al-Yaseen, Othman, & Nazri, 2017) used simple machine learning algorithms for intrusion detection and also gained sufficient detection accuracy with considerable FAR. The main advantage with these methods is that they have less computational time and burden because they used simple algorithms and not focused over the optimization algorithms. However, they are not robust to some datasets which have complex traffic connections with unknown features. For instance, the traditional ELM won't use any kernel at the classification process. Without the utilization of a kernel, the ML algorithm makes the system weak in learning and cannot provide a generalization capability. In such case, they perform well for known attacks and shown poor performance for minor attacks. Next, the ANN algorithm is susceptible for non-linear features, as the features of normal and attack connections. Even though the *k*-NN algorithm very simple but and robust for complex connections, because the nearest neighbors are chosen based in the computation of Euclidean distances between features.

Table 7. Performance comparison of proposed method with recent methods

Method	Average Accuracy (%)	Average FAR (%)
GHSOM + NSGA-II (De la Hoz et al., 2014)	99.1200	2.2400
OS-ELM + FST (Singh et al., 2015)	98.6600	1.7400
PSOM + PCA + FDR (De la Hoz et al., 2015)	90.0000	NA
BC + *k*-NN (Li et al., n.d.)	94.9200	1.5900
SVM + ANN (Hussain et al., 2016)	93.2300	NA
FC-ANN (Wang et al., 2010)	96.5000	NA
PSO + GA + ACO + REPT (Tama et al., 2019)	91.2700	8.9000
K-means + SVM + ELM (Al-Yaseen, Othman, & Nazri, 2017)	95.7500	NA
Spark ML + Conv-LSTM (Khan et al., 2019)	97.2900	0.7100
MRFS-MC-SVM (Veeranna & Reddy, n.d.)	95.6345	0.7664
Proposed	**98.9856**	**0.6112**

FC-ANN (Wang et al., 2010) employed fuzzy clustering before training the system and employed an ANN model for every cluster individually. In such case, the ANN performs well because the connections in one cluster are linearly related. Hence its accuracy is better than the methods without clustering. Next, (Khan et al., 2019) applied both ML and deep learning for the detection intrusions. However, the deep learning is not suitable for text data mining because, it works based on the convolutions. Moreover, they applied spark ML (combination of different ML algorithms like SVM, DT, RF and Gradient boosting) which creates a huge computational burden over the system. Next, OS-ELM is an online method that applied an alpha profiling to lessen the time complexity by discarding the redundant features through an ensemble filter, consistency and correlation based feature selection technique. Due to this online mechanism, this approach is achieved considerable accuracy along with less FAR.

In our earlier contribution (Veeranna & Reddy, n.d.), we mainly focused on the removal of duplicate features and hence we have gained only 95.6345% accuracy. However, in the current proposed method, we focused on the removal of duplicate connections along with duplicate features. Moreover, we opt two simple and effective ML algorithms those make the system with string learning ability and good generalization capability. Due to these advantages, we have gained 98.9856% accuracy with a small FAR of 0.6112%. Compared with all the existing methods, the proposed method has more detection performance with less computational complexity.

Table 8 shows the comparison of computational time for proposed and different existing hybrid models. The proposed approach has gained very less computational time at both training and testing phases, because the proposed MRFS mechanism finds only 33 features for all classes. As the feature count is less, the time required to train and test will get reduced and hence it has obtained less time. Among the two existing methods, even though (Tama et al., 2019) has reduced the feature count, the computation time is high because they employed three optimization algorithms whose main drawback is huge time complexity. Next, the method proposed in (Hussain et al., 2016) didn't apply any feature extraction thus it considered an entire 41 features for classification. Even though it was employed entire 41 features, the time complexity is less compared to (Tama et al., 2019) because, the time complexity of ANN and SVM is less compared to optimization algorithms (PSO, GA and ACO). ML algorithms are static algorithms but optimization algorithms are iterative algorithms. The varying characteristics of training time for different features are shown in Figure 6.

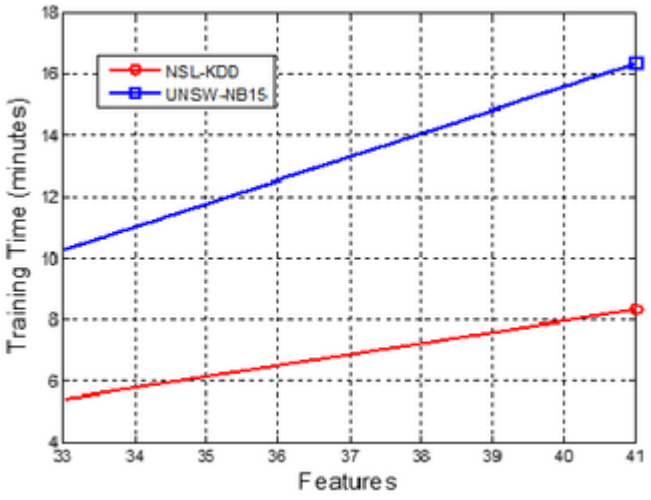
5. CONCLUSION

This paper modeled a hybrid IDS mechanism whose main objective is to deter the intrusion from network traffic. The proposed model is an integrated form of two machine learning algorithms they are CELM and SVM. CELM is an extended version of ELM and applied as a second stage classifier for misuse detection. SVM is used a first stage classifier to spate normal activities from malicious activities. Further, to get optimal results at training phase, we employed data pre-processing, followed by clustering, redundant connections removal and redundant features removal. Experiments are

Table 8. Average time for training and testing processes for NSL-KDD dataset

Method	Training Time (min)	Testing Time (min)	Features
SVM + ANN (Hussain et al., 2016)	8.3325	6.1478	41
PSO + GA + ACO + REPT (Tama et al., 2019)	15.6345	10.2214	37
Proposed	5.4127	4.0023	33

Figure 6. Training time for different features



investigated through NSL-KDD and UNSW-NB15 datasets and the performance is assessed through several performance metrics. The observed accuracy, FAR and computational time shows the superiority of proposed method than the existing methods.

Since there is a huge rise in new and uneven attacks patterns, the feature extraction based on internal characteristics of traffic connections are need to be considered during the feature extraction. Further, this work can also be enhanced by introducing a customized deep learning model at classification.

REFERENCES

Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372. doi:10.1016/j.asoc.2015.10.011

Al-Jarrah, O. Y., Alhussein, O., Yoo, P. D., Muhaidat, S., Taha, K., & Kim, K. (2015). Data randomization and cluster-based partitioning for botnet intrusion detection. *IEEE Transactions on Cybernetics*, *46*(8), 1796–1806. doi:10.1109/TCYB.2015.2490802 PMID:26540724

Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. *Expert Systems with Applications*, *67*, 296–303. doi:10.1016/j.eswa.2016.09.041

Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. A. (2017). Real-time multi-agent system for an adaptive intrusion detection system. *Pattern Recognition Letters*, *85*, 56–64. doi:10.1016/j.patrec.2016.11.018

Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time varying chaos particle swarm optimization. *Neurocomputing*, *199*, 90–102. doi:10.1016/j.neucom.2016.03.031

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2-3), 191–203. doi:10.1016/0098-3004(84)90020-7

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proc. 5th Annual Workshop on Computational Learning Theory*, 144-152. doi:10.1145/130385.130401

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials*, *18*(2), 1153–1176. doi:10.1109/COMST.2015.2494502

Canbay, Y., & Sagiroglu, S. (2016). A hybrid method for intrusion detection. *IEEE International Conference on Machine Learning and Applications*, 156–161.

Chang & Lin. (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*. 2(3). Software available at https://www.csie.ntu.edu.tw/~cjlin/libsvm

De la Hoz, E., De la Hoz, E., Ortiz, A., Ortega, J., & Martínez-Álvarez, A. (2014). Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems*, *71*, 322–338. doi:10.1016/j.knosys.2014.08.013

De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., & Prieto, B. (2015). PCA filtering and probabilistic SOM for network intrusion detection. *Neurocomputing*, *164*, 71–81. doi:10.1016/j.neucom.2014.09.083

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. doi:10.1109/4235.996017

Du, M., Wang, K., Chen, Y., Wang, X., & Sun, Y. (2018). Big data privacy preserving in multi- access edge computing for heterogeneous internet of things. *IEEE Communications Magazine*, *56*(8), 62–67. doi:10.1109/MCOM.2018.1701148

Du, M., Wang, K., Xia, Z., & Zhang, Y. (2018). Differential privacy preserving of training model in wireless big data with edge computing. *IEEE Transactions on Big Data*.

Eesa, Brifcani, & Orman. (2014). A New Tool for Global Optimization Problems-Cuttlefish Algorithm", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:8, No:9.

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670–2679. doi:10.1016/j.eswa.2014.11.009

Feng, X., Xiao, Z., Zhong, B., Qiu, J., & Dong, Y. (2018). Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing*, *65*, 139–151. doi:10.1016/j.asoc.2018.01.021

Folino, G., & Sabatino, P. (2016). Ensemble based collaborative and distributed intrusion detection systems: A survey. *Journal of Network and Computer Applications*, *66*, 1–16. doi:10.1016/j.jnca.2016.03.011

Gaikwad, D., & Thool, R. C. (2015). Intrusion detection system using bagging ensemble method of machine learning. *2015 International Conference on Computing Communication Control and Automation*, 291–295. doi:10.1109/ICCUBEA.2015.61

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feed forward neural networks. *2004 IEEE Int. Jt. Conf. Neural Networks*, 1–4, 985–990. doi:10.1109/IJCNN.2004.1380068

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1-3), 489–501. doi:10.1016/j.neucom.2005.12.126

Hubballi, N., & Suryanarayanan, V. (2014). False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, *49*, 1–17. doi:10.1016/j.comcom.2014.04.012

Hussain, J., Lalmuanawma, S., & Chhakchhuak, L. (2016). A two-stage hybrid classification technique for network intrusion detection system. *International Journal of Computational Intelligence Systems*, *9*(5), 863–875. doi:10.1080/18756891.2016.1237186

Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K. K., & Anwar, S. (2016). Intrusion response systems: Foundations, design, and challenges. *Journal of Network and Computer Applications*, *62*, 53–74. doi:10.1016/j.jnca.2015.12.006

Ji, S. Y., Jeong, B. K., Choi, S., & Jeong, D. H. (2016). A multi-level intrusion detection method for abnormal network behaviors. *Journal of Network and Computer Applications*, *62*, 9–17. doi:10.1016/j.jnca.2015.12.004

Joldzic, O., Djuric, Z., & Vuletic, P. (2016). A transparent and scalable anomaly-based dos detection method. *Computer Networks*, *104*, 27–42. doi:10.1016/j.comnet.2016.05.004

Khan, M. A., Karim, M., & Kim, Y. (2019). A scalable and hybrid intrusion detection system based on the convolutional-lstm network. *Symmetry*, *11*(4), 583. doi:10.3390/sym11040583

Khraisat, Gondal, Vamplew, & Kamruzzaman. (2020). Survey of intrusion detection systems: techniques, datasets and challenges. *Cyber Security*.

Khraisat, A., Gondal, I., & Vamplew, P. (2018). An anomaly intrusion detection system using C5 decision tree classifier,. *Proc. Pacific Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer. doi:10.1007/978-3-030-04503-6_14

Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing*, *18*, 178–184. doi:10.1016/j.asoc.2014.01.028

Li, Yu, Bai, Hou, & Chen. (n.d.). An effective two-step intrusion detection approach based on binary classification and k-NN", IEEE Access, Volume: 6, pp. 12060 - 12073

Mishra, Varadharajan, Tupakula, & Pilli. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutor.

Moustafa, N., & Slay, J. (2015). *Unsw-nb15: A Comprehensive Data Set for Network Intrusion Detection Systems (Unsw-nb15 Network Data Set). In 2015 military communications and information systems conference (MilCIS)*. IEEE.

Nguyen, H., Franke, K., & Petrovic, S. (2010). Improving effectiveness of intrusion detection by correlation feature selection. *5th Int. Conf. Availability, Reliab. Secur.*, 17–24. doi:10.1109/ARES.2010.70

Peddabachigari, S., Abraham, A., Grosan, C., & Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications*, *30*(1), 114–132. doi:10.1016/j.jnca.2005.06.003

Pham, N. T., Foo, E., Suriadi, S., Jeffrey, H., & Lahza, H. F. M. (2018). Improving performance of intrusion detection system using ensemble methods and feature selection. *Proceedings of the Australasian Computer Science Week Multiconference*. ACM. doi:10.1145/3167918.3167951

Rauber, A., Merkl, D., & Dittenbach, M. (2002). The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, *13*(6), 1331–1341. doi:10.1109/TNN.2002.804221 PMID:18244531

Roulston, M. S. (1999). Estimating the errors on measured entropy and mutual information. *Physica D. Nonlinear Phenomena*, *125*(3), 285–294. doi:10.1016/S0167-2789(98)00269-3

Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. *Computer Networks*, *148*, 164–175. doi:10.1016/j.comnet.2018.11.010

Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. *Computer Networks*, *148*, 164–175. doi:10.1016/j.comnet.2018.11.010

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, *31*(3), 357–374. doi:10.1016/j.cose.2011.12.012

Singh, R., Kumar, H., & Singla, R. K. (2015). An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Systems with Applications*, *42*(22), 8609–8624. doi:10.1016/j.eswa.2015.07.015

Smits, G. F., & Jordaan, E. M. (2002). Improved SVM regression using mixtures of kernels. *Proceeding 2002 Int. Jt. Conf. Neural Networks*, *1–3*, 2785–2790.

Tama, B. A., Comuzzi, M., & Rhee, K. H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 94497–94507. doi:10.1109/ACCESS.2019.2928048

Tao, P. Y., Sun, Z., & Sun, Z. X. (2018). An improved intrusion detection algorithm based on GA and SVM. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 13624–13631. doi:10.1109/ACCESS.2018.2810198

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD cup 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6. doi:10.1109/CISDA.2009.5356528

Tian, Y., Mirzabagheri, M., Bamakan, S. M. H., Wang, H., & Qu, Q. (2018, October). Ramp loss one-class support vector machine; A robust and effective approach to anomaly detection problems. *Neurocomputing*, *310*, 223–235. doi:10.1016/j.neucom.2018.05.027

Tian, Z. D., Li, S. J., Wang, Y. H., & Wang, X. D. (2018). Wind power prediction method based on hybrid kernel function support vector machine. *Wind Engineering*, *42*(3), 252–264. doi:10.1177/0309524X17737337

Tsai, Hsu, Lin, & Lin. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications*, vol. 36, no. 10, pp. 11 994 – 12 000.

Veeranna & Reddy. (n.d.). Sliding Window Assisted Mutual Redundancy Based Feature Selection for Intrusion Detection System. International Journal of Ad Hoc and Ubiquitous Computing.

Villalba, L. J. G., Orozco, A. L. S., Vidal, J. M., Member, S., Orozco, A. L. S., & Vidal, J. M. (2015). Anomaly-based network intrusion detection system. *IEEE Lat. Am. Trans.*, *13*, 850–855. doi:10.1109/TLA.2015.7069114

Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, *37*(9), 6225–6232. doi:10.1016/j.eswa.2010.02.102

Wang, K., Du, M., Sun, Y., Vinel, A., & Zhang, Y. (2016). Attack detection and distributed forensics in machine-to-machine networks. *IEEE Network*, *30*(6), 49–55. doi:10.1109/MNET.2016.1600113NM

Wu, J., Zhu, Y., Wang, Z. C., Song, Z. J., Liu, X. G., Wang, W. H., Zhang, Z. Y., Yu, Y. S., Xu, Z. P., Zhang, T. J., & Zhou, J. H. (2017). A novel ship classification approach for high resolution SAR images based on the BDA-KELM classification model. *International Journal of Remote Sensing*, *38*(23), 6457–6476. doi:10.1080/01431161.2017.1356487

Thotakura Veeranna obtained his Bachelor's in Computer Science and Information Technology from the Jawaharlal Nehru Technological University, Hyderabad and Master's in Computer Science and Engineering from the Jawaharlal Nehru Technological University, Hyderabad. Currently, he is a research scholar in the Jawaharlal Nehru Technological University Kakinada, Kakinada, Andhra Pradesh. He is a member of ISTE. He has 14 years of teaching experience. His research interest area is data mining and network security. Currently, he is working as an Assistant Professor in the CSE Department at Sai Spurthi Institute of Technology, B. Gangaram, Telangana. He published four international journal and one conference.

Kiran Kumar Reddi obtained his Masters degree in JNTU, Kakinada and Doctor of Philosophy in Computer Science and Engineering from the Acharya Nagarjuna University, Guntur. Currently, he is working as an Assistant Professor in the Department of Computer Science at Krishna University, Machilipatnam, Andhra Pradesh. He is a member of ISTE, CSI and IETE. He is specialised in data mining, bioinformatics, network security, and cloud computing. He published ten international journals and four international conferences and one national conference. He is guiding more than ten research scholars as a supervisor and co-supervisor from JNTUK, Kakinada.