Network Intrusion Detection using Fusion Features and Convolutional Bidirectional Recurrent Neural Network

Jagruthi H[a], Manjunath Mulimani[b,*], Ritika Nandi[b], Bheemappa Halavar[c], Kavitha C[1]

[a]*Department of Information Science and Engineering, BNM Institute of Technology, Bangalore, 560 070, India*
[b]*Department of Computer Science and Engineering, MIT, Manipal Academy of Higher Education, Manipal, 576 104, India*
[c]*Department of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, 560 054, India*

Abstract

In this paper, a novel fusion features to train Convolutional Bidirectional Recurrent Neural Network (CBRNN) are proposed for network intrusion detection. UNSW-NB15 dataset's attack behaviors (input features) are fused with their first and second-order derivatives at different stages to get fusion features. In this work, we have taken architectural advantage and combine both Convolutional Neural Network (CNN) and bidirectional Long Sort-Term Memory (LSTM) as Recurrent Neural Network (RNN) to get CBRNN. The input features and their first and second-order derivatives are fused and considered as input to CNN and this fusion is known as early fusion. Outputs of the CNN layers are fused and used as input to the bidirectional LSTM, this fusion is also known as late fusion. Results show that late fusion features are more suitable for intrusion detection and achieve good performance.

Keywords: Intrusion detection, fusion features, Convolutional Neural Network (CNN), bidirectional Long Sort-Term Memory (LSTM), Convolutional Bidirectional Recurrent Neural Network (CBRNN), UNSW-NB15 dataset

*Corresponding author

Email addresses: jagruthi.aries@gmail.com, Contact No.: +91-872-246-7250 (Jagruthi H), manjunath.gec@gmail.com, Contact No.: +91-974-207-3368 (Manjunath Mulimani), ritika.nandi77@gmail.com (Ritika Nandi), bheemhh@msrit.edu (Bheemappa Halavar), kavitha_prasanna@yahoo.com (Kavitha C)

1. Introduction

In recent years the number of internet users is increased rapidly. Emerging sectors such as social media, education, tourism, banking and so on are now connected by www and reachable through the internet. Rapid usage of services provided by the above sectors generates a massive amount of data over the internet. Cyber attacks are also increasing quickly with the growth of the internet. Attack on valuable data may break confidentiality, integrity and availability of policies of computer security [1].

There are several free hacking attacking tools widely present on the internet and their usage does not require any high skills [2]. On the other hand, multiple protection software tools against attacks are also available. Some of the protection software tools are encryption methods, antivirus, firewalls, etc [3]. However, these software tools are not effectively preventing all types of threats [4]. Hacking of client's passwords, getting or modifying client's sensitive information leads to loss of credibility of business providers with their clients.

The Intrusion Detection System (IDS) is a primary security mechanism that continuously monitors and filters out the network activities affected by the attack. IDS is widely used for the detection of various types of attacks. IDSs are broadly classified into two types, one is network-based and another one is host-based IDS [5]. Network-based IDS detects malicious activities in network traffic by analyzing individual packets. Host-based IDS detects the malicious activities from logs. Combination of network and host-based IDS are widely used in various organization. However traditional methods show low performance in the detection of unseen threats. In this work, the main concentration is on network-based IDS.

In the literature different machine learning algorithms such as Naive-Bayes (NB) [6], Logistic Regression (LR) [7], Gradient Boosting (GB) [8], Random Forest (RF), K-Nearest Neighbour (KNN) [9] and so on are used for intrusion detection. Different feature selection algorithms are used to select significant

features from the data and used as input to the machine learning algorithms [4]. Recently, deep learning models such as Convolutional Neural Network (CNN) [10], Recurrent Neural Network (RNN) [11] are also used for intrusion detection. DNN models omit feature selection stage and data itself considered as input features.

In this paper, architectures of CNN and RNN are combined to get Convolutional Recurrent Neural Network (CRNN). CNN captures significant features from each feature vector and RNN captures sequential information from the input features. CRNN takes the advantages of both the architectures. Bidirectional Long Short-Term Memory (LSTM) is used as RNN. Hence, the proposed model is also referred to as Convolutional Bidirectional Recurrent Neural Network (CBRNN). Input features and their first and second-order derivatives are fused at different levels to train CBRNN.

Rest of the paper is organized as follows. Section 2 gives in detail the proposed method for intrusion detection. Section 3 introduces evaluation methods. Section 4 discuses the results and section 5 concludes the work.

2. Literature Review

In this section well-established intrusion detection system reported in the literature are described below. Intrusion detection is a machine learning problem that includes three main stages. The first one is a selection of input datasets, the second one is feature selection and finally, classification. Related work reported in the literature on each stage is described below in brief. Different datasets are used to evaluate the performance of the network intrusion detection system in the literature. However, KDD99 and UNSWNB-15 are the most widely used datasets [12][13]. Recently, KDD99 is enhanced and developed a new dataset namely NSLKDD [14]. The following list of problems associated with the NSLKDD.

- The training and testing set includes repeated observations(features). Hence, classifies bias more towards frequent observations.

- It is an imbalanced dataset each class includes imbalanced observations and it decreases the overall performance.

UNSWNB-15 dataset includes several advantages as compared to NSLKDD and they are listed below.

- It has a similar probability distribution for both testing and training datasets.
- UNSWNB-15 includes complex patterns. Intrusion detection algorithms on UNSWNB-15 achieve less performance accuracy when compared to NSLKDD.

Hence, UNSWNB-15 dataset is more challenging and suitable to evaluate the performance of the proposed methodology.

There are thwo feature selection approaches reported in the literature: filter and wrapper [15]. The filter approach heuristic approach that uses statistical measures to eliminate redundant features and selects significant features for the classifier [16]. Some of the techniques used in the filter approach are Principal Component Analysis (PCA), chi-square, information gain, and correlation coefficient [17]. Whereas the wrapper approach uses the classifier itself to select the features which give the best results [18]. The techniques used here are genetic algorithms, recursive feature elimination, sequential search [19]. Wrapper-based approaches are expensive as compared to filter-based approaches when dealing with large datasets with too many features [20]. A significant subset of features selected by the feature selection algorithm used to train a classifier for intrusion detection.

The different traditional statistical classifiers are used for intrusion detection in the literature. Recently, Deep learning techniques such as CNN and RNN are widely used for image and speech recognition [21][22]. Very few works reported in the literature which use either CNN or RNN for intrusion detection. As mentioned earlier, DNN omits the separate feature selection stage and selects the significant features in different layers of DNN. In this work, CNN and RNN are combined for effective IDS.

3. Network intrusion detection

The proposed network intrusion detection system consists of two stages. One is fusion feature generation and another is the design of CBRNN architecture. Each stage is explained below in brief.

3.1. Generation of fusion features

The input dataset of network intrusion is a combination of different features of the form discrete, continuous and categorical values with varying ranges and resolution. Even a few features of the dataset may be null or infinite. Current machine learning/deep learning algorithms are not compatible with such data types. Hence, the dataset is preprocessed first to get significant features suitable for classification. Further, first and second-order derivates are computed from preprocessed features of size $M \times N$ using (1) and (2).

$$d_m = \begin{cases} c_{m+1} - c_m & \text{if } m = 1 \\ c_m - c_{m-1} & \text{otherwise} \end{cases} \tag{1}$$

$$a_m = \begin{cases} d_{m+1} - d_m & \text{if } m = 1 \\ d_m - d_{m-1} & \text{otherwise} \end{cases} \tag{2}$$

The first (d_m) and second-order (a_m) derivative features are also known as delta and delta-delta (acceleration) features respectively and widely used as supplementary features in speech/speaker recognition tasks [23]. Preprocessed features and their first and second-order derivatives are fused (concatenated) to get fusion features.

3.2. Convolutional Bidirectional Recurrent Neural Network

Two CBRNN architectures of the proposed work are given in Fig. 1 and 2. Both the architectures consist of one 1D CNN layer followed by Relu activation function and 1D max-pooling operation. The output of the CNN layer is fed as input to a bidirectional LSTM layer, the output of the bidirectional LSTM is

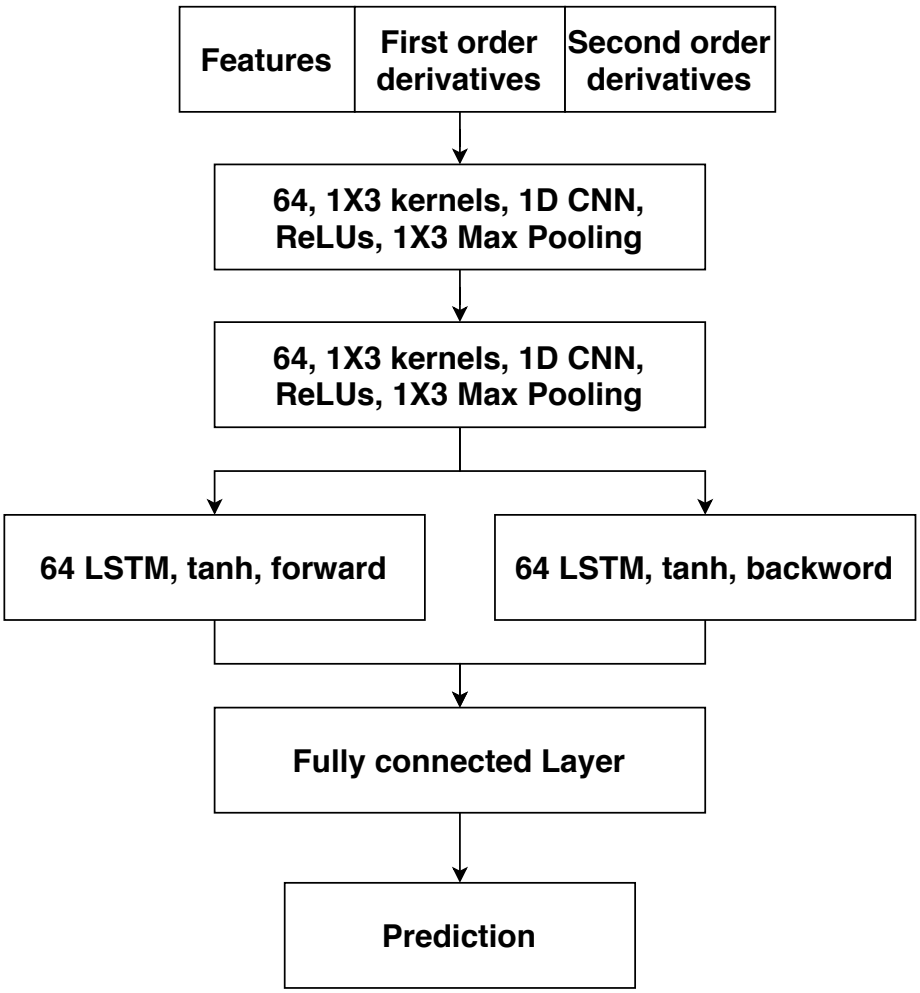

Fig. 1. Convolutional Bidirectional Recurrent Neural Network for early fusion features.

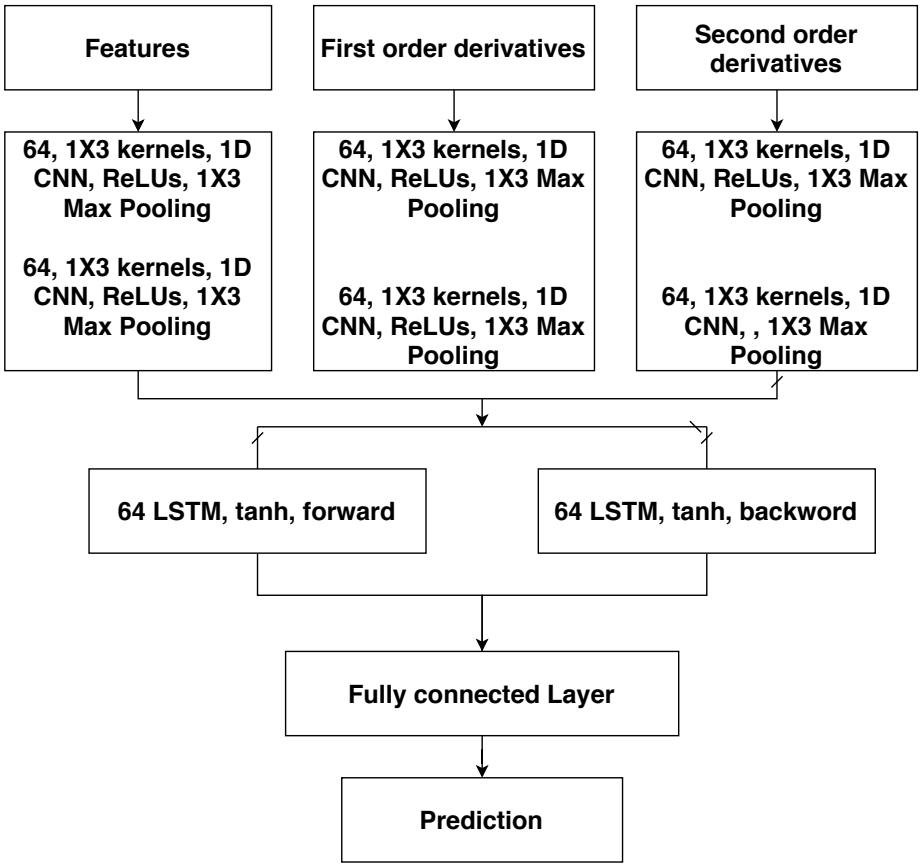

Fig. 2. Convolutional Bidirectional Recurrent Neural Network for late fusion features.

input to the fully connected output layer with a sigmoid or softmax activation function. Each layer of CBRNN architecture is explained below in brief.

3.2.1. CNN layer

In this work, there are two ways are proposed to consider input to the CNN: preprocessed features and their first and second-order statistics are fused to get features of size $M \times N \times 3$ and fed as input to the CNN of CBRNN. This fusion is also known as early fusion (see Fig. 1). In another case, preprocessed features and their first and second-order statistics of sizes $M \times N \times 1$ (where M and N are the number of rows and columns respectively) are fed as input to the three separate CNNs. The outputs of these CNNs of size $M \times N' \times K$ (where N' is the number of columns remaining after convolution and pooling operations, K is the number of kernels) are fused and fed as input to the two side-by-side bidirectional LSTM (see Fig. 2). This fusion is also known as late fusion.

Kernels of the CNN layer are the 1D sequences and convolved over row direction (feature vector) of the feature matrix. Similarly, 1D non-overlapping max-pooling is used to reduce the dimension of each feature vector of X. CNN and max-pooling applied only along N dimension, hence the number of feature vectors M is unaffected.

3.2.2. bidirectional LSTM and output layers

Bidirectional LSTM is an RNN with two hidden layers side-by-side. The input feature and its reverse are considered as input to the first (forward) and second (backward) units respectively. Forward and backward units together provide additional information to the network to learn effectively. Hence, this approach is referred to as CBRNN (Convolutional Bi-directional Recurrent Neural Network). The output layer is the fully connected feedforward layer. The number of hidden units in the output layer is the same as the number of intrusion classes in the dataset. A sigmoid/softmax activation function is used in the output layer to predict classes of each feature vector.

4. Evaluation

4.1. Dataset

The performance of the proposed methodology is evaluated on UNSW-NB15 dataset [13]. It includes 47 features and two class labels. One class label represents a category of attack and another represents the state of attack. The category of attack denotes ten categorical labels. Out of ten, nine denotes categories of attacks, namely, analysis, backdoor, DoS (Denial-of-Service), exploits, fuzzers, generic, reconnaissance, shellcode, worms and one label denotes normal (non-attack). The state of the attack denotes two categorical labels: normal and attack. Anyone of the class label (either category of attack or state of the attack) is used for classification. The dataset is divided into training and testing sets. 10% of the training set is used as a validation set. There are only 42 features are present in the training/testing sets. These features are categorized into five groups, namely, basic, content, flow, time and additional generated features.

To evaluate the performance of the proposed methodology, we used the class labels: category of attack and state of the attack separately. If the class label is a category of attack then intrusion detection on UNSW-NB15 is a multi-class prediction problem, otherwise, it is a binary class prediction problem with the state of the attack class label.

4.2. Evaluation metrics

There are three metrics, namely, accuracy, Detection Rate (DR) and False Alarm Rate (FAR) are reported in the literature for intrusion detection [24][25]. Accuracy metric is adopted in this work to evaluate the performance of the proposed methodology.

4.3. Neural network configuration

The hyperparameters of the proposed CBRNN give the best results on the validation set are selected to predict the classes of the test set. The number of

hidden units in both CNN and bidirectional LSTM is set to 64. The size of the kernels in the CNN layer is 1×3. Each CNN followed by max-pooling operation of size 1×3 only along N direction. Binary cross-entropy and categorical cross-entropy are used as loss functions for binary and multi-class classification respectively. Adam is used as a gradient descent optimizer to predict an attack. The proposed network is trained using the backpropagation algorithm according to the values of the loss function obtained through each successive iteration. The proposed CBRNN network is implemented using the Tensorflow library of Python.

4.4. Performance comparison

To evaluate the performance of the proposed CBRNN, we compare it with CNN and bidirectional LSTM alone. CNN architecture is obtained by replacing bidirectional LSTM and RNN is obtained by removing CNN from CBRNN. For both CNN and LSTM, we use the same set of hyperparameters described for CBRNN.

5. Results and Discussion

Experimental results are reported in this section. All the reported results are obtained from the test set of UNSW-NB15. Each experiment is repeated ten times and their average is reported. In this work, we perform two series of experiments. One is the prediction of binary classes (normal or attack) and the other is the prediction of multiple categories of attack, which are explained below in brief.

The UNSW-NB15 features are normalized to zero mean and unit variance. These normalized features are considered as input to the CNN, bidirectional LSTM (BRNN) and CBRNN. The performance of each architecture is tabulated in Table 1. It is observed that CNN learns from important information in each feature vector effectively than BRNN. CBRNN combines significant information from both CNN and BRNN and consistently outperforms the CNN and BRNN

Table 1: Comparison of overall Recognition accuracy (%) of proposed CBRNN with other architectures for binary classification on UNSW-NB15 dataset.

Method	Accuracy
CNN	88.23
BRNN	85.73
CBRNN	90.04

Table 2: Comparison of overall Recognition accuracy (%) of proposed CBRNN using early and late fusion methods for binary classification on UNSW-NB15 dataset.

Fusion	Feature combinations	Accuracy
Early	UNSW-NB15 features + FOD	90.49
	UNSW-NB15 features + FOD + SOD	90.72
Late	UNSW-NB15 features-CNN + FOD-CNN	91.89
	UNSW-NB15 features-CNN + FOD-CNN + SOD-CNN	92.13

FOD: First-order derivatives, SOD: Second-order derivatives.

alone. Normalized features of UNSW-NB15 are fused with their first and second-order derivatives. The results of early and late fusion features are reported in the Table 2. It is observed that both early and late fusion features improve the performance of CBRNN. However, late fusion features relatively perform better than early fusion features, which are generated by simple concatenation.

Experimental results for recognition of multiple categories of attack are tabulated in Table 3. As expected, late fusion features perform better than early fusion features. However, the performance of CBRNN for recognition of multiple categories of attack is reduced as compared recognition of binary attack or normal classes. Since UNSW-NB15 is the highly imbalanced dataset. Each attack includes a different number of instances. Hence, few attacks such as *analysis*, *worms*, *backdoor* and *shellcode* include very few training instances and they are may not sufficient to train CBRNN. Hence, they are excluded from training and testing data and Table 3 shows result on the remaining data. The performance of the proposed method is also compared with machine learning methods reported in the literature on UNSW-NB15 such as Naive Bayes (NB) [6], Gradient Boosting (GB) [8], KNN and J48 [9] in Fig. 3. However, UNSW-NB15 is a recent data and very few works are reported on this dataset in the

Table 3: Comparison of overall Recognition accuracy (%) of proposed CBRNN using early and late fusion methods for multiple attack categories classification on UNSW-NB15 dataset.

Fusion	Feature combinations	Accuracy
Early	UNSW-NB15 features + FOD	75.08
	UNSW-NB15 features + FOD + SOD	75.90
Late	UNSW-NB15 features-CNN + FOD-CNN	77.12
	UNSW-NB15 features-CNN + FOD-CNN + SOD-CNN	78.55

FOD: First-order derivatives, SOD: Second-order derivatives.

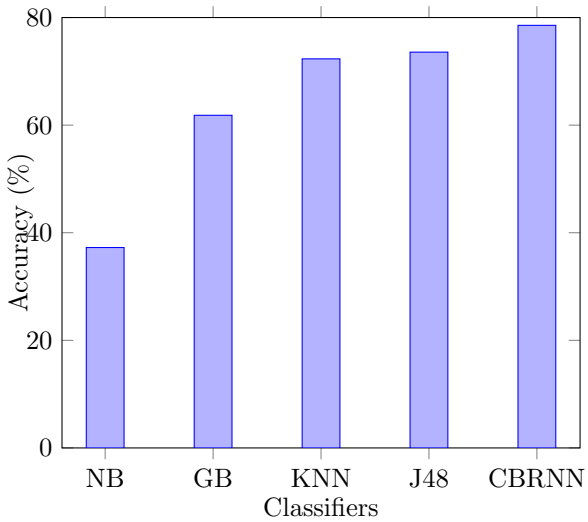

Fig. 3. Performance of the proposed approach concerning different machine learning classifiers.

literature. The reported works use a different proportion of data, a different number of categories of attack, different preprocessing techniques, and different feature selection methods. In most cases, these details are not provided. Hence, the comparison with other existing methods is difficult. However, the comparison shown in Fig. 3 is just for reference and is not claimed that proposed CBRNN outperform the other methods. The proposed method not used any feature selection methods, CNN identifies the significant features using its convolutional and max-pooling operations and achieves good performance.

6. Conclusions

In this paper, a novel fusion features + CBRNN are proposed for intrusion detection. The outputs of individual CNN layers are fused to get late fusion features, which are effective than simple concatenated early fusion features. Results show that late fusion features + CBRNN outperform early fusion features. Combined CBRNN outperforms the individual CNN and bidirectional LSTM. Late fusion features + CBRNN achieve overall 92.13% and 78.55% recognition accuracy for binary and multi-class classification. It indicates that proposed late fusion features + CBRNN have a significant contribution towards the network intrusion detection. In the future, the use of different feature selection methods may further improve the performance of the proposed approach.

References

[1] W. Lee, S. J. Stolfo, A framework for constructing features and models for intrusion detection systems, ACM transactions on Information and system security (TiSSEC) 3 (4) (2000) 227–261.

[2] T. A. Tchakoucht, M. Ezziyyani, Building a fast intrusion detection system for high-speed-networks: probe and dos attacks detection, Procedia Computer Science 127 (2018) 521–530.

[3] Y. Y. Chung, N. Wahid, A hybrid network intrusion detection system using simplified swarm optimization (sso), Applied soft computing 12 (9) (2012) 3014–3022.

[4] C. Khammassi, S. Krichen, A ga-lr wrapper approach for feature selection in network intrusion detection, computers & security 70 (2017) 255–277.

[5] V. Hajisalem, S. Babaie, A hybrid intrusion detection system based on abc-afs algorithm for misuse and anomaly detection, Computer Networks 136 (2018) 37–50.

[6] N. Moustafa, J. Slay, The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems, in: 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS), IEEE, 2015, pp. 25–31.

[7] S. Bagui, E. Kalaimannan, S. Bagui, D. Nandi, A. Pinto, Using machine learning techniques to identify rare cyber-attacks on the unsw-nb15 dataset, Security and Privacy 2 (6) (2019) e91.

[8] S. Meftah, T. Rachidi, N. Assem, Network based intrusion detection using the unsw-nb15 dataset, International Journal of Computing and Digital Systems 8 (5) (2019) 478–487.

[9] M. H. Kamarudin, C. Maple, T. Watson, N. S. Safa, A logitboost-based algorithm for detecting known and unknown web attacks, IEEE Access 5 (2017) 26190–26200.

[10] R. Vinayakumar, K. Soman, P. Poornachandran, Applying convolutional neural network for network intrusion detection, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 1222–1228.

[11] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access 5 (2017) 21954–21961.

[12] C. Elkan, Results of the kdd'99 classifier learning, Acm Sigkdd Explorations Newsletter 1 (2) (2000) 63–64.

[13] N. Moustafa, J. Slay, 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, 2015, pp. 1–6.

[14] L. Dhanabal, S. Shantharajah, A study on nsl-kdd dataset for intrusion detection system based on classification algorithms, International Journal

of Advanced Research in Computer and Communication Engineering 4 (6) (2015) 446–452.

[15] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, N. Yazdani, Mutual information-based feature selection for intrusion detection systems, Journal of Network and Computer Applications 34 (4) (2011) 1184–1199.

[16] S. Solorio-Fernández, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, A new hybrid filter–wrapper feature selection method for clustering based on ranking, Neurocomputing 214 (2016) 866–880.

[17] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the 20th international conference on machine learning (ICML-03), 2003, pp. 856–863.

[18] O. Depren, M. Topallar, E. Anarim, M. K. Ciliz, An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks, Expert systems with Applications 29 (4) (2005) 713–722.

[19] I. Guyon, S. Gunn, M. Nikravesh, L. A. Zadeh, Feature extraction: foundations and applications, Vol. 207, Springer, 2008.

[20] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, B. Prieto, Pca filtering and probabilistic som for network intrusion detection, Neurocomputing 164 (2015) 71–81.

[21] M. Mulimani, S. G. Koolagudi, Locality-constrained linear coding based fused visual features for robust acoustic event classification, in: Proceedings of INTERSPEECH 2019, 2019, pp. 2558–2562.

[22] A. K. Kamath, A. T. Karthik, L. Monis, M. Mulimani, S. G. Koolagudi, Sobriety testing based on thermal infrared images using convolutional neural networks, in: TENCON IEEE Region 10 Conference, IEEE, 2018, pp. 2170–2174.

[23] M. Mulimani, S. G. Koolagudi, Segmentation and characterization of acoustic event spectrograms using singular value decomposition, Expert Systems with Applications 120 (2019) 413–425.

[24] W.-C. Lin, S.-W. Ke, C.-F. Tsai, Cann: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowledge-based systems 78 (2015) 13–21.

[25] N. Moustafa, J. Slay, The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set, Information Security Journal: A Global Perspective 25 (1-3) (2016) 18–31.