An Intrusion Detection System based on Deep Belief Networks

Othmane Belarbi[1], Aftab Khan[*1], Pietro Carnelli[1], and Theodoros Spyridopoulos[2]

[1] Toshiba Europe Ltd., Bristol Research & Innovation Laboratory, Bristol, UK
{othmane.belarbi, aftab.khan, pietro.carnelli}@toshiba-bril.com
[2] Cardiff University, School of Computer Science & Informatics, Cardiff, UK
spyridopoulost@cardiff.ac.uk

Abstract. The rapid growth of connected devices has led to the proliferation of novel cyber-security threats known as zero-day attacks. Traditional behaviour-based Intrusion Detection Systems (IDSs) rely on Deep Neural Networks (DNNs) to detect these attacks. The quality of the dataset used to train the DNNs plays a critical role in the detection performance, with underrepresented samples causing poor performances. In this paper, we develop and evaluate the performance of Deep Belief Networks (DBNs) on detecting cyber-attacks within a network of connected devices. The CICIDS2017 dataset was used to train and evaluate the performance of our proposed DBN approach. Several class balancing techniques were applied and evaluated. Lastly, we compare our approach against a conventional Multi-Layer Perceptron (MLP) model and the existing state-of-the-art. Our proposed DBN approach shows competitive and promising results, with significant performance improvement on the detection of attacks underrepresented in the training dataset.

Keywords: Network Intrusion Detection System · Deep Learning · Deep Belief Networks · Class Balancing

1 Introduction

Traditional Intrusion Detection Systems (IDSs) are signature-based, relying on existing signatures of known attacks. They are designed to detect single attacks or families of attacks based on their particular characteristics. However, the vast amount of data that needs to be daily processed to develop new signatures renders their timely generation a challenging task. Furthermore, the generation of signatures for complex attacks that have evolved from previously known threats is not straightforward and requires additional effort [32].

Behaviour analysis techniques have also been explored by researchers to detect intrusions [32]. Most of these systems use machine learning techniques to model system behaviour and detect malicious activity. Compared to signature-based IDSs, machine learning techniques can detect large families of attack variants regardless of their complexity. Typical machine learning techniques for

classification are used to train models of malicious behaviour based on existing labelled datasets of system activity [1,18]. Classification can be binary (i.e. normal or malicious) or multi-class. In multi-class classification, the model can also detect the type of attack based on the classes/labels in the training set.

The quality of the dataset plays a critical role in the detection performance of the trained IDS model. When a type of attack is underrepresented in the dataset, typical in IDS datasets, the resulting model performs poorly on the detection of attack variants that belong to the infrequent attack type. Several attempts have been proposed to mitigate the issues caused by imbalanced IDS datasets, focusing mainly on the data sampling and class balancing techniques [25].

In this paper, we propose a multi-class classification Network Intrusion Detection System (NIDS) based on Deep Belief Networks (DBNs)[3]. DBN is a generative graphical model formed by stacking multiple Restricted Boltzmann Machines (RBMs). It can identify and learn high-dimensional representations. A DBN is first pre-trained in an unsupervised way by using a greedy layer-by-layer learning algorithm and then fine-tuned using the back-propagation technique in a supervised manner. As shown in our experimental results, this two-stage training process improves the detection performance against infrequent attack samples whilst retaining a high performance against the rest of the attacks.

This paper's contributions can be summarised as follows:

1. We demonstrate that our DBN-based NIDS outperforms MLP-based NIDSs, especially when there is a small number of attack samples in the dataset.
2. We conducted a series of class balancing experiments on the highly imbalanced CICIDS2017 dataset [26], which includes benign network traffic and twelve attacks. Our experimental results demonstrate that our class balancing approach improves the detection performance in terms of F1-score.
3. The classification results of our NIDS, after applying our class balancing approach, are compared against the state-of-the-art. Our proposed method demonstrates significant improvement in F1-score from 0.873 to 0.94.

The rest of the paper is organised as follows. In Section 2, we present recent related work on IDSs and DBNs. Section 3 describes the proposed methodology for the DBN-based NIDS. In Section 4, we analyse the pre-processing approach we followed on the CICIDS2017 dataset and the model architecture we used, and report the performance results of the study comparing them against the state-of-the-art. Finally, we conclude the paper and present pathways for future work in Section 5.

2 Related Work

In general, all ML-based IDSs comprise multiple components including data collection, pre-processing, feature extraction/selection and decision engine [5]. NIDS rely on capturing and analysing inbound and outbound network traffic

[3] https://github.com/othmbela/dbn-based-nids

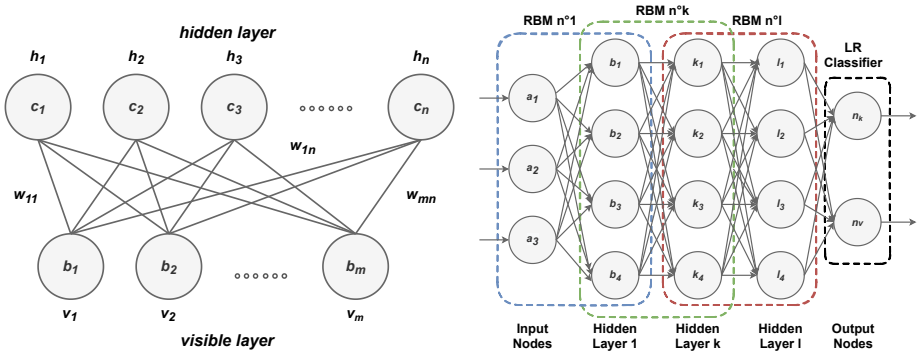
in the under analysis system. In most cases, they are separate devices attached to the network gateway. Depending on the application and the related privacy policy, they may collect and analyse only the headers of the network traffic packets or require access to the packet payload as well [19]. Our work focuses on the pre-processing, the feature extraction/selection and the decision engine for the implementation of NIDS. Our NIDS model relies on the CICIDS2017 dataset [26], which was generated by analysing full-packet network traffic. However, the extracted data and our generated features do not rely on the payload of the network packets nor on the source/destination IP address and port number. Therefore, our model preserves user privacy to an extent. Nevertheless, user privacy largely depends on the way data are captured on the network, which is outside the scope of our research.

Various machine learning methodologies have been researched for the development of NIDS [4,13,17]. Among these, Support Vector Machines (SVMs) have been one of the most commonly used [7,11,12]. Researchers in [7] combined SVMs with ensemble learning to increase the generalisation ability of the model and improve its performance on the unknown data samples. However, both the Fuzzy-C Means (FCM) clustering method and the k-fold cross validation method used to train and test the final SVM can be problematic when applied to network traffic time series datasets. A similar issue is evident in other current SVM-based NIDS works [11]. Additionally, research conducted in [3] demonstrated that SVMs can take a significantly longer time to classify unseen data.

Recent work has demonstrated that deep learning techniques are quite efficient in identifying cyber-attacks on networks [27]. In [31], the authors proposed a deep learning approach for intrusion detection using Recurrent Neural Networks (RNNs). The proposed RNN-IDS effectively recognised the type of intrusion with a higher accuracy and detection rate than traditional ML-based IDS in both binary and multi-class classification. However, it performed poorly against the minority classes in multi-class classification, a common issue in cyber-security datasets. In a similar work [21], it was demonstrated that an ensemble of RNNs can perform binary classification with 94% accuracy within the first 5 seconds of execution. However, no discussion was provided by the authors around the performance of the approach against the minority classes of imbalanced datasets.

In [22], the authors implemented and tested four classifiers for Distributed Denial Of Service (DDoS) attack detection using deep learning models such as MLP, 1D-Convolutional Neural Network (CNN), LSTM and 1D-CNN+LSTM. The models were built based on the CICIDS2017 dataset [26]. The dataset was balanced by using the duplicating method. The experimental results demonstrated that the CNN+LSTM performed better than the rest of the deep learning models achieving an accuracy of 97.16%.

The researchers in [8] presented a deep learning detection system for DDoS attacks, called LUCID. LUCID is a CNN-based IDS for binary classification of DDoS attacks. The proposed approach can be used in online resource-constrained environments thanks to its ability to ensure low processing overhead and attack



(a) RBM (b) DBN

Fig. 1: Algorithm structure

detection time. The evaluation results showed that the CNN-based IDS recognises DDoS attack on the CICIDS2017 dataset with an accuracy of 99.7%.

The authors in [9] focused on real time detection of malicious traffic in large-scale high-throughput networks. The proposed approach, named Whisper, leverages frequency domain analysis to extract and analyse sequential information of the network traffic. Whisper can effectively detect 42 sophisticated attacks from the WIDE MAWI dataset in high throughput with at least 90% detection accuracy. Even though our work does not focus on large-scale networks, combining it with frequency domain analysis is an interesting concept we would like to explore in future work.

Researchers in [29] used an RBM to extract fundamental features from the traffic data and then train an SVM classifier. This approach was able to shorten the training time whilst preserving performance of the classifier. In a similar work [20], a NIDS based on deep learning was built using the KDD CUP'99 dataset. The authors used an RBM to extract higher-level features from the input data followed by a back-propagation neural network to classify intrusions. The suggested model showed significant improvement in accuracy over traditional ML-based NIDS. Finally, a system that classifies network intrusions using DBNs (a stack of RBMs) was implemented in [2]. In that work, the DBN-IDS achieved an accuracy of around 97.5% using 40% of the dataset in the training process. However, the evaluation metrics were only limited to accuracy, with no discussion around recall and precision. The proven ability of RBMs to improve the performance of existing classifiers, along with the lack of an in-depth analysis of DBNs as NIDS and the limited work on tackling imbalanced cyber-security datasets, inspired the research presented in this paper.

2.1 Restricted Boltzmann Machines

RBMs are Energy-Based Models (EBMs) that have been largely used for several tasks such as feature extraction, feature reduction and collaborative filtering

[15,23]. RBMs are two-layer undirected models where the layers are the visible layer and the hidden layer as illustrated in Fig. 1a.

An RBM captures the dependency between the visible and hidden units, v and h, by associating an energy to each configuration of the variables. The energy function takes low values when the two variables are compatible and higher values when h is less compatible with v.

The energy function of a joint configuration (v, h) has an energy given by [16]:

$$E(v,h) = -\sum_{i \in visible} b_i v_i - \sum_{j \in hidden} c_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{1}$$

where:

- v_i, h_j are the binary states of visible unit i and hidden unit j.
- b_i, c_j are the biases of visible unit i and hidden unit j.
- w_{ij} is the weight between the visible unit i and hidden unit j.

The joint probability distribution of a pair of a visible vector v and a hidden vector h is defined via the energy function as:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{2}$$

where Z, the partition function, is defined as the summation over all possible pairs of visible and hidden vectors:

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{3}$$

The probability of the visible vector v is given by marginalising out the hidden vector h:

$$p(v) = \frac{1}{Z} \sum_{h} e^{-E(v,h)} \tag{4}$$

During the training phase, RBMs adjust their weights based on the Contrastive Divergence (CD) algorithm in which the second expectation term of the gradient descent, Equation 5, is approximated after running k steps of the Gibbs sampler.

$$\Delta w_{ij} = \epsilon \big(< v_i h_j >_{data} - < v_i h_j >_{recon} \big) \tag{5}$$

where:

- ϵ is the learning rate.
- $< v_i h_j >_{data}$ is the product of v_i, h_j before reconstruction
- $< v_i h_j >_{recon}$ is the expected value of v_i, h_j after k-step reconstruction

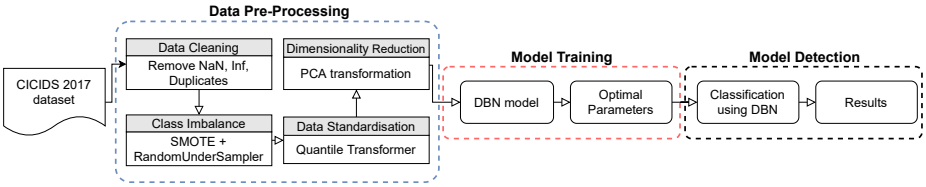


Fig. 2: High-level architecture of our DBN-based NIDS

2.2 Deep Belief Network

A DBN is a deep architecture of multiple stacks of RBMs sequentially connected as shown in Fig. 1b. Each RBM model performs a non-linear transformation on its input vectors and produces outputs vectors that will serve as input for the next RBM model in the sequence. Except for the first and final layers of the DBN, every layer serves as a hidden layer to the nodes that come before and as a visible/input layer to the nodes that comes after. DBNs can be used in both unsupervised learning for generating images and in supervised learning for classification [24]. DBNs are pre-trained layer by layer and fine-tuned using the back-propagation technique.

3 Methodology

In this paper, we aim to evaluate DBNs for network intrusion detection and address the issue of high-class imbalance in network traffic datasets. Fig. 2 illustrates the high level architecture of our approach. In the next sections we present and analyse the key processes in our architecture.

3.1 Training Data Pre-Processing

We split the dataset into training, validation, and testing sets. The training set is used to train the model; the model learns the mapping function from this data. The validation set is used to perform the initial testing and tuning of the model, while the testing set is used to evaluate the model's performance. In datasets with high-class imbalance (as is generally observed in IDS datasets), a stratified split of the data is very important allowing a fairer assessment of the model's performance [14]. Stratified data splitting preserves the proportion of samples for each target class essentially ensuring that enough samples of the minority class are included in the training set.

Class Imbalance: As highlighted above, IDS datasets generally suffer from higher class imbalance compared to datasets from other domains. For example, datasets such as NSL-KDD and CICIDS2017 have 95% and 90% benign samples respectively, with the rest containing different types of attacks. This can fundamentally be attributed to the nature of these attacks, even in real-world deployments where the majority of the network traffic is expected to be benign

in nature. High-class imbalance in datasets introduces bias in favour of the majority class (benign), which makes the classification of the minority classes very challenging. There are multiple methods introduced in the literature that can handle such class imbalance. We evaluate the following methods in this paper:

Under-sampling and Over-sampling: Under-sampling methods re-sample the *majority* class by removing some of the instances. Typically, a random under-sampler is employed in which the majority class samples are removed at random. Over-sampling methods on the other hand re-sample the *minority* classes by duplicating or creating new artificial instances with algorithms such as the Synthetic Minority Over-sampling Technique (SMOTE) [6].

Class Weight Strategy: Models could consider the underrepresented classes without re-sampling the training set by adding weights to the loss function of the algorithm [28]. The class weight strategy penalises the wrong classification of the minority classes more than the misclassification of the majority class. Hence, each class will have equal importance on gradient updates, on average, regardless of the number of samples.

Sample Weight Strategy: Another method consists of associating a weight to each training sample where the weight is computed as described above. In this strategy, the dataset is re-balanced by ensuring that each batch of data is proportionally distributed.

Data Standardisation: IDS datasets are sometimes high dimensional in nature (e.g. CICIDS2017 dataset has 78 features). In the case of such higher-dimensional datasets without any form of normalization, machine learning models in general do not optimise very well or take much longer to train. Standardising the input variables to overcome this issue can become skewed or biased, due to the large number of outliers in the highly imbalanced security datasets. Dropping outliers may help, however, it is not recommended since there is a risk of losing essential information required for correctly classifying attacks. There are different methods to overcome this problem by using statistics that are robust to outliers. Unlike other scalers, robust scaler and quantile transformation methods are not influenced by the impact of marginal outliers (because they are based on percentiles and quantiles respectively). In robust scaling, the median of a given feature is subtracted from values and divided by the interquartile range. The resulting range is larger than the other scalers. The quantile transformation is another standardization method that is robust to outliers. This method estimates the cumulative distribution function of the input variables and transforms the values to a uniform distribution [0, 1]. Then, the obtained values are mapped to the desired distribution using the associated quantile function.

Dimensionality Reduction: One of the most important steps in typical machine learning toolchains is dimensionality reduction. This can be achieved through feature selection or other methods such as Principal Components Analysis (PCA). PCA is a data analysis technique that transforms the possible correlated features set to uncorrelated features set. It can be used to reduce the dimensionality feature space by ignoring the components containing the least variance of the original feature space.

Table 1: Class distribution of CICIDS2017 dataset.

Category	**Labels**	**# samples**
Benign	Benign	1,807,787
DoS/DDoS	Heartbleed, DDoS DoS Hulk, DoS GoldenEye DoS Slowloris, DoS Slowhttptest	320,269
PortScan	PortScan	57,305
Brute Force	FTP-Patator, SSH-Patator	8,551
Web Attack	Web Attack – Brute Force Web Attack – XSS Web Attack – SQL Injection	2,118
Botnet	Bot	1,943

3.2 Model Training

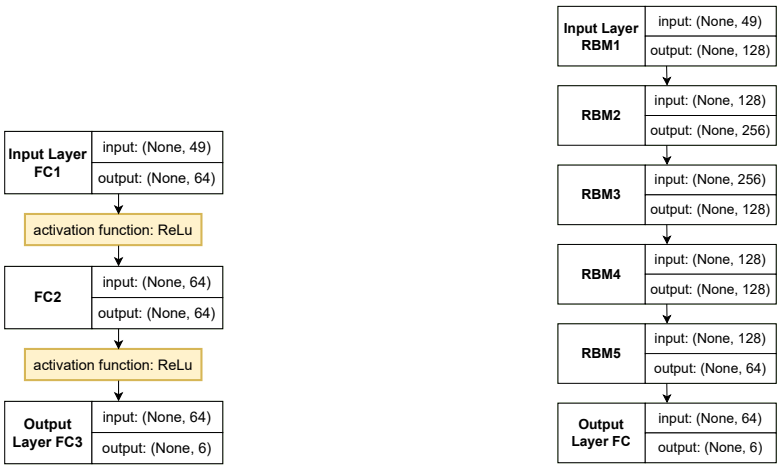
The DBN model requires two steps in the training process: unsupervised pre-training and supervised fine-tuning. In the first phase, each RBM is trained to reconstruct its input by adjusting its weights and feeding the input layer of the next RBM. This process is repeated until each RBM layer is pre-trained (greedy learning). In supervised fine-tuning, all the weights are optimised by using the stochastic gradient descent and back-propagation. Details of the specific architectures, hyperparameters, and performance metrics are provided in Section 4. Finally, in the case of intrusion datasets that mainly suffer from high-class imbalance, the accuracy metric cannot be relied upon [10]. Therefore, we chose the F1-score, precision, and recall as our metrics to evaluate the models.

4 Experiments

All the experiments were conducted using a 64-bit Intel(R) Core(TM) i7-7500U CPU with 16GB RAM in Windows 10 environment. The models have been implemented in Python v3.8.2 using the PyTorch v1.9.0 library.

4.1 Dataset

In order to test the proposed DBN-based NIDS, we used the CICIDS2017 dataset created by the Canadian Institute for Cyber-security (CIC) which satisfies the eleven criteria described in [26]. The dataset includes samples of benign activity and common attacks over a period of five days. A B-Profile system described in [26] was used to generate realistic benign traffic. The benign traffic corresponds to the human interaction of 25 users based on standard network protocols such as HTTP(S), FTP, SSH, IMAP and POP3. Several attacks were then initiated.



(a) MLP architecture (b) DBN architecture

Fig. 3: Model architecture

Each record in the dataset has 78 network features extracted with a network traffic analyser, CICFlowMeter [26].

We removed ten features that had no variance and seventeen highly correlated features from the dataset since they were bringing redundant information to the model. A detailed description of our pre-processing can be found on the github repository[3]. The dataset was then split into training, validation, and testing sets in the proportions of 60%, 20%, and 20% respectively. The CICIDS2017 dataset contains many outliers. Therefore, we used the quantile transformation to deal with outliers and transform our features to a uniform distribution [0, 1].

Table 1 shows how we created new attack classes by merging minority classes that had similar characteristics and behaviour. After merging the classes, we explored different class balancing techniques to further reduce the class imbalance rate and improve the prevalence ratio of the dataset. The "Infiltration" class was removed because of its small portion of the overall dataset.

We then performed a PCA transformation on the CICIDS2017 dataset to reduce the dimensional feature space and simplify the learning process. As stated in the previous section, it is important to preserve as much information as possible. After a series of experiments we concluded that 99% of the explained variance can be retained for only 25 principal components. The feature space has been almost halved without losing any significant information.

4.2 MLP and DBN Architecture

In this study, we have implemented and tuned two different deep learning classifiers, MLP and DBN. Fig. 3a shows the architecture of the implemented MLP. It consists of multiple fully connected layers. 49 nodes were used in the input layer

Table 2: Model design and parameters.

(a) DBN

Parameter	Pre-training	Fine-tuning
Epochs	10	30
Learning rate	0.1	0.001
Batch size	64	128
Momentum	0.9	-
Optimiser	SGD	Adam
Loss function	-	cross-entropy
Gibbs step	1 step	-
Weight init.	Xavier initialiser	-
Bias init.	Zeros (0)	-

(b) MLP

Parameter	
Epochs	10
Learning rate	0.02
Batch size	64
Momentum	0.9
Weight decay	-
Optimiser	SGD
Loss function	cross-entropy

to represent the number of input features. After fine-tuning, two hidden layers with 64 nodes each were set and the ReLU activation function was used in the hidden layers. Six nodes were used in the output layer, each node representing one class. Finally, the Soft-Max function was used in the output layer to perform a multi-class classification.

Fig. 3b shows the architecture of the implemented DBN. After fine-tuning, five RBMs are stacked with (49, 128), (128, 256), (256, 128), (128, 128), and (128, 64) visible/hidden nodes set per RBM respectively. The output from the last RBM is connected to a fully connected layer with 6 nodes for multi-class classification using the Softmax function. The training parameters of the DBN and MLP are shown in Table 2.

4.3 Results

We compare the performance of the different class balancing techniques using the evaluation metrics defined in Section 3.2.

Fig. 4 shows the F1-score, precision, and recall that the DBN-based NIDS achieved for various class balancing techniques. The model has a high precision and recall, thus a high F1-score, for the "Benigh", "Brute Force", "DoS/DDoS", and "PortScan" classes. However, for the "Web Attack" and "Botnet" classes, either high recall and low precision, or vice versa is observed. It may not always be possible to have high values for both precision and recall. Hence, a trade-off between the precision and the recall is required. In the context of cyber-security, it is important to be able to detect all the malicious activities on the network (high recall). However, it is less severe if few false alarms are raised (low precision). Fig. 4 shows that the combination of SMOTE and random under-sampler achieved the best results. The recall value of every class is at least equal to 99%, which means that most of the attacks are successfully detected. Moreover, this method has the best precision for the Botnet class in comparison to the other class balancing techniques.

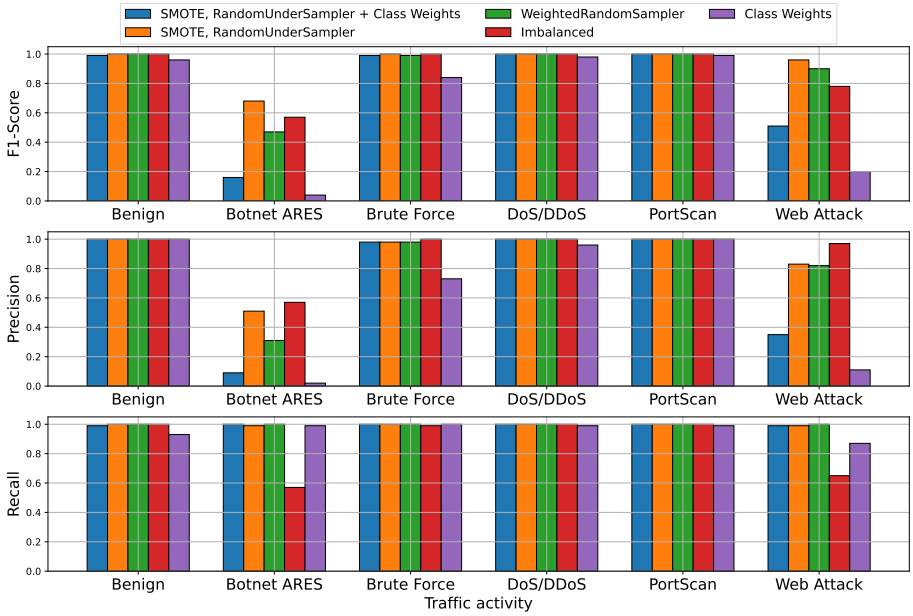


Fig. 4: F1-Score, Precision and Recall achieved by the proposed DBN-based multi-class classification for the different balancing techniques

Table 3 shows the confusion matrix of the DBN and MLP on the testing set. The number of misclassifications and correct classification are summarised with count values for each label. We can see that both models can correctly classify most of the network traffic samples. However, significant differences can be seen between these two models in terms of precision. The MLP classifies precisely only 49% of the "Web Attack"; 407 benign traffic packets were misclassified as "Web Attack". On the other hand, the DBN model classifies "Web Attacks" with a precision of 83%. This suggests that the DBN model has developed a more meaningful pattern for these attacks, potentially due to its two-stage training.

We also compare our DBN and MLP models against the state-of-the-art intrusion detection approaches that use the CICIDS2017 dataset. All studied approaches trained their models using supervised learning methods. However, some of the studies perform binary rather than multi-class classification. We compare our approach against six methods. DeepGFL [30] is a framework that can extract deep features from attributed network flow graphs. LSTM and 1D-CNN [22] are deep learning models that perform binary classification. 1D-CNN+LSTM [22] is a combination of the two latter models for binary classification. LUCID [8] uses CNNs to detect DDoS attacks. Table 4 summarises the performance results of the aforementioned models and compares them against our models.

The DeepGFL [30] framework used for the classification of twelve classes produced worse results than our proposed approach in terms of recall and F1-

Table 3: Confusion matrices.

(a) MLP

Actual	**Predicted**						**Recall**
	Benign	Botnet	Brute Force	DoS/DDoS	PortScan	Web Attack	
Benign	360810	500	29	101	7	407	100%
Botnet	6	381	0	0	0	0	98%
Brute Force	4	0	1689	0	0	0	100%
DoS/DDoS	113	0	0	63717	0	17	100%
PortScan	3	4	1	23	11366	4	100%
Web Attack	2	0	0	2	0	408	99%
Precision	100%	43%	98%	100%	100%	49%	

(b) DBN

Actual	**Predicted**						**Recall**
	Benign	Botnet	Brute Force	DoS/DDoS	PortScan	Web Attack	
Benign	361350	358	28	52	6	60	100%
Botnet	3	384	0	0	0	0	99%
Brute Force	3	0	1691	0	0	0	100%
DoS/DDoS	119	0	0	63707	0	21	100%
PortScan	6	4	0	17	11371	3	100%
Web Attack	5	0	0	1	0	406	99%
Precision	100%	51%	98%	100%	100%	83%	

score. The DeepGFL approach classified correctly only 44.8% of the malicious activities. The differences in the results could be explained either by the nature of the model itself or the different pre-processing technique used.

Furthermore, our DBN approach achieved higher classification results than the MLP, 1D-CNN and LSTM in [22]. Although these classifiers were designed for the detection of DDoS attacks, our approach performed better while classifying six different types of attacks. For instance, the MLP proposed by [22] and our MLP achieve the same F1-score but differences can be seen in terms of precision and recall. Although our MLP presents a slightly smaller precision there has been a major improvement in the recall. As mentioned in Section 4.3, we prefer higher recall over precision since it is fundamental to be able to detect all malicious activities on computer networks even if it comes at the expense of a few false positives. Finally, LUCID [8] and 1D-CNN+LSTM [22] performed slightly better than our DBN approach. However, these two models were only used for binary classification whereas our DBN approach achieved a high recall while performing a multi-class classification.

Table 4: Performance comparison against existing methods using the CICIDS2017 dataset, the target number of classes is shown in the last column.

Study	**Method**	**F1-score**	**Recall**	**Precision**	**#Classes**
Our Study	DBN	0.940	0.997	0.887	6
Our Study	MLP	0.873	0.995	0.817	6
[30]	DeepGFL	0.531	0.448	0.948	12
[22]	MLP	0.872	0.862	0.884	2
[22]	LSTM	0.895	0.898	0.984	2
[22]	1D-CNN	0.939	0.901	0.981	2
[22]	1D-CNN+LSTM	0.982	0.991	0.974	2
[8]	LUCID	0.996	0.999	0.993	2

5 Conclusion and Future Work

In this paper, we researched the use of DBNs for Network Intrusion Detection. We developed two NIDS based on DBNs and MLPs. We conducted multiple experiments using the CICIDS2017 dataset with various class-balancing techniques. The proposed MLP-based and DBN-based NIDS achieved an F1-score of 87.3% and 94% respectively using a combination of SMOTE and random under-sampler. Our experimental results demonstrate that DBNs surpass traditional MLPs on the classification of network intrusions, especially when the attacks have a small number of samples. Moreover, we compared our proposed DBN-based NIDS against the state-of-the-art ML-based IDS methods. Our NIDS outperforms all other multi-class classification approaches and shows competitive results against binary classification methods with a major improvement in terms of recall.

Currently, our method requires a centralised network sampler in order to monitor network traffic for our DBN-based NIDS. However, in certain applications this may not be possible due to increased data volumes, relevant privacy issues or complex network configurations. As such, a distributed approach might be necessary. We are currently working towards this direction exploring the various capabilities of DBNs when deployed in a distributed manner.

References

1. Al-Qatf, M., Lasheng, Y., Al-Habib, M., Al-Sabahi, K.: Deep learning approach combining sparse autoencoder with svm for network intrusion detection. IEEE Access **6**, 52843–52856 (2018). https://doi.org/10.1109/ACCESS.2018.2869577
2. Alom, M.Z., Bontupalli, V., Taha, T.M.: Intrusion detection using deep belief networks. In: National Aerospace and Electronics Conference (NAECON). pp. 339–344 (2015). https://doi.org/10.1109/NAECON.2015.7443094
3. Anthi, E., Williams, L., Słowińska, M., Theodorakopoulos, G., Burnap, P.: A supervised intrusion detection system for smart home iot devices. IEEE Internet of Things Journal **6**(5), 9042–9053 (2019). https://doi.org/10.1109/JIOT.2019.2926365

4. Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L.: Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences **378**, 484–497 (2017). https://doi.org/10.1016/j.ins.2016.04.019
5. Bridges, R.A., Glass-Vanderlan, T.R., Iannacone, M.D., Vincent, M.S., Chen, Q.G.: A survey of intrusion detection systems leveraging host data. ACM Comput. Surv. **52**(6) (nov 2019). https://doi.org/10.1145/3344382
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (Jun 2002). https://doi.org/10.1613/jair.953
7. Chitrakar, R., Huang, C.: Selection of candidate support vectors in incremental svm for network intrusion detection. Computers & Security **45**, 231–241 (2014). https://doi.org/10.1016/j.cose.2014.06.006
8. Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martínez-del Rincón, J., Siracusa, D.: Lucid: A practical, lightweight deep learning solution for ddos attack detection. IEEE Transactions on Network and Service Management **17**(2), 876–889 (2020). https://doi.org/10.1109/TNSM.2020.2971776
9. Fu, C., Li, Q., Shen, M., Xu, K.: Realtime robust malicious traffic detection via frequency domain analysis. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (Nov 2021). https://doi.org/10.1145/3460120.3484585
10. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: Bagging, boosting, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(4), 463–484 (2012). https://doi.org/10.1109/TSMCC.2011.2161285
11. Gu, J., Lu, S.: An effective intrusion detection approach using svm with naïve bayes feature embedding. Computers & Security **103**, 102158 (2021). https://doi.org/10.1016/j.cose.2020.102158
12. Gu, J., Wang, L., Wang, H., Wang, S.: A novel approach to intrusion detection using svm ensemble with feature augmentation. Computers & Security **86**, 53–62 (2019). https://doi.org/10.1016/j.cose.2019.05.022
13. Guo, C., Ping, Y., Liu, N., Luo, S.S.: A two-level hybrid approach for intrusion detection. Neurocomputing **214**, 391–400 (2016). https://doi.org/10.1016/j.neucom.2016.06.021
14. Hammerla, N.Y., Plötz, T.: Let's (not) stick together: Pairwise similarity biases cross-validation in activity recognition. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing. p. 1041–1051. Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2750858.2807551
15. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006). https://doi.org/10.1126/science.1127647
16. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (Aug 2002). https://doi.org/10.1162/089976602760128018
17. Kim, G., Lee, S., Kim, S.: A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with Applications **41**(4, Part 2), 1690–1700 (2014). https://doi.org/10.1016/j.eswa.2013.08.066
18. Kunang, Y.N., Nurmaini, S., Stiawan, D., Suprapto, B.Y.: Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. Journal of Information Security and Applications **58**, 102804 (2021). https://doi.org/10.1016/j.jisa.2021.102804

19. Matyás, V., Kur, J.: Conflicts between intrusion detection and privacy mechanisms for wireless sensor networks. IEEE Security Privacy **11**(5), 73–76 (2013). https://doi.org/10.1109/MSP.2013.111
20. Peng, W., Kong, X., Peng, G., Li, X., Wang, Z.: Network intrusion detection based on deep learning. In: 2019 International Conference on Communications, Information System and Computer Engineering (CISCE). pp. 431–435 (2019). https://doi.org/10.1109/CISCE.2019.00102
21. Rhode, M., Burnap, P., Jones, K.: Early-stage malware prediction using recurrent neural networks. Computers & Security **77**, 578–594 (2018). https://doi.org/10.1016/j.cose.2018.05.010
22. Roopak, M., Yun Tian, G., Chambers, J.: Deep learning models for cyber security in iot networks. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0452–0457 (2019). https://doi.org/10.1109/CCWC.2019.8666588
23. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning. p. 791–798. ICML '07 (2007). https://doi.org/10.1145/1273496.1273596
24. Salama, M.A., Hassanien, A.E., Fahmy, A.A.: Deep belief network for clustering and classification of a continuous data. In: The 10th IEEE International Symposium on Signal Processing and Information Technology. pp. 473–477 (2010). https://doi.org/10.1109/ISSPIT.2010.5711759
25. Sapre, S., Islam, K., Ahmadi, P.: A comprehensive data sampling analysis applied to the classification of rare iot network intrusion types. In: 2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC). pp. 1–2 (2021). https://doi.org/10.1109/CCNC49032.2021.9369617
26. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications (2018). https://doi.org/10.5220/0006639801080116
27. Singla, A., Bertino, E., Verma, D.: Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In: Asia Conference on Computer and Communications Security (2020). https://doi.org/10.1145/3320269.3384718
28. Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P.J.: Training deep neural networks on imbalanced data sets. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 4368–4374 (2016). https://doi.org/10.1109/IJCNN.2016.7727770
29. Yang, J., Deng, J., Li, S., Hao, Y.: Improved traffic detection with support vector machine based on restricted boltzmann machine. Soft Comput. **21**(11), 3101–3112 (Jun 2017). https://doi.org/10.1007/s00500-015-1994-9
30. Yao, Y., Su, L., Lu, Z.: Deepgfl: Deep feature learning via graph for attack detection on flow-based network traffic. In: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM). pp. 579–584 (2018). https://doi.org/10.1109/MILCOM.2018.8599821
31. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954–21961 (2017). https://doi.org/10.1109/ACCESS.2017.2762418
32. Zhong, W., Yu, N., Ai, C.: Applying big data based deep learning system to intrusion detection. Big Data Mining and Analytics **3**(3), 181–195 (2020). https://doi.org/10.26599/BDMA.2020.9020003