TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 4, August 2021, pp. 1407~1418
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i4.18772

Deep learning with focal loss approach for attacks classification

Yesi Novaria Kunang[1], Siti Nurmaini[2], Deris Stiawan[3], Bhakti Yudho Suprapto[4]
[1]Doctoral Engineering Departement, Faculty of Engineering, Universitas Sriwijaya, Indonesia
[1,2]Intelligent System Research Group, Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[3]Computer Networking & Information Systems, Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[1]Computer Science Departement, Universitas Bina Darma, Indonesia
[4]Electrical Engineering Departement, Universitas Srwijaya, Indonesia

Article Info

Article history:

Received Jun 20, 2020
Revised Jun 7, 2021
Accepted Jun 17, 2021

Keywords:

Attack classification
Focal loss
Imbalanced data
Intrusion detection system
Multi-class

ABSTRACT

The rapid development of deep learning improves the detection and classification of attacks on intrusion detection systems. However, the unbalanced data issue increases the complexity of the architecture model. This study proposes a novel deep learning model to overcome the problem of classifying multi-class attacks. The deep learning model consists of two stages. The pre-tuning stage uses automatic feature extraction with a deep autoencoder. The second stage is fine-tuning using deep neural network classifiers with fully connected layers. To reduce imbalanced class data, the feature extraction was implemented using the deep autoencoder and improved focal loss function in the classifier. The model was evaluated using 3 loss functions, including cross-entropy, weighted cross-entropy, and focal losses. The results could correct the class imbalance in deep learning-based classifications. Attack classification was achieved using automatic extraction with the focal loss on the CSE-CIC-IDS2018 dataset is a high-quality classifier with 98.38% precision, 98.27% sensitivity, and 99.82% specificity.

Corresponding Author:

Siti Nurmaini
Intelligent System Research Group, Faculty of Computer Science Department
Universitas Sriwijaya
Fakultas Ilmu Komputer, Srijaya Negara Bukit Besar St. Palembang 30139
Email: sitinurmaini@gmail.com

1. INTRODUCTION

The increasing use of the internet of things technology has significantly increased the amount of data generated daily. Apart from providing the benefits and facilities that improve the quality of life, big data and internet of things (IoT) pose the challenge of vulnerability and security issues [1]-[4]. Vulnerability increased due to the large number of physical infrastructures connected to the network [5], [6]. To address this issue, the network intrusion detection system (NIDS) is an appropriate alternative for securing modern networks [7].

The security solutions of NIDS-based IoT have been proposed, such as machine and deep learning-based methods [7]-[10]. deep learning (DL) has impressively performed in big data domains [11], including computer vision [12], [12] speech recognition [13], and medical imaging [14]-[16]. Also, some studies have used the DL algorithm on the intrusion detection system, such as deep Boltzmann machines [17], deep belief networks [18], and deep neural networks [19], [20]. The DL technique or its combination improves the accuracy and detection speed [21]. Furthermore, it efficiently detects attack variants and patterns [22]. There are several challenges to NIDS in efficiently and effectively detecting the network’s abnormal behavior. IoT applications' big data nature presents difficulties with the amount and complexity of data [1]. Many applications encrypt Internet traffic using security protocols. The encryption encourages more diverse and sophisticated attacks that need to be detected. Second, in real network traffics, data distribution is usually imbalanced [23]. In this case, the number of samples belonging to the regular traffic is much higher than the attack sample. Therefore, the classification results lean towards the benign data [24].

Most of the datasets used to analyze NIDS, such as network security layer- knowledge discovery in database (NSL-KDD) [25], the communications security establishment and the Canadian institute for cybersecurity intrusion detection system 2018 dataset (CSE-CIC-IDS2018) [26], have a severe class imbalance problem. Several approaches have been employed to solve this problem, such as over-sampling [23], [27], [28], under-sampling [29], spread sub-sample [30], and class balancer [23]. However, the fundamental problem of class imbalance in attack classes remains an interesting issue to study. Also, data quality issues trigger imbalanced data problems [31], [32]. Therefore, other strategies are needed to solve this problem, especially for multi-class cases. There have been developments in the fields of focal loss in image recognition, biomedical sciences, and stability training [33]-[35]. Using this knowledge, the improved focal loss function for a multi-class model is used to prevent class imbalance and over-fitting attack classification. This research focuses on efficient training on all data sets, based on the extreme class imbalance. Moreover, the training is based on multi-class attacks by utilizing the focal loss function used in the deep learning models.

This study proposes a multi-class focal loss function of deep learning to address unbalanced data. The result is compared with the cross-entropy (CE) loss and weighted cross-entropy functions. As a contribution, this study proposes the deep auto-encoder (DAE), combined with deep neural network (DNN) model using a multi-class focal loss function. This is aimed to address the different class imbalance for the attack classification. Experimental results show that the pre-training stage, deep auto-encoder, has advantages in more complicated features learned from the original data. Focal loss function, scaled from cross-entropy loss, is a more effective alternative to previous approaches in dealing with the class imbalance in multi-class attack classification.

2. RELATED WORK

Network Intrusion Detection System has been studied widely over the past several years. This section briefly discusses some published approaches to deep learning methods, in particular to imbalanced datasets. In 2019, Lin *et al.* [26] used deep learning for dynamic network anomaly detection. The synthetic minority oversampling technique (SMOTE) algorithm was experimentally applied to handle the imbalanced class problem in the CSE-CICIDS2018 dataset. As a classifier, a deep neural network model was used with long short-term memory (LSTM) based, combined with an attention mechanism (AM), to enhance performance. The SMOTE algorithm applied to promote the proportion of minority class optimizes the deep learning model. The model achieved the best results, with an accuracy of 96.2%, and the recall rate reached 98% for 6 categories class.

More recently, Zhang *et al.* [27] introduced a hybrid SMOTE that combines SMOTE and Gaussian mixture model (GMM) based clustering to improve the minority class's detection rate. The synthetic minority over-sampling technique (SMOTE) and gaussian mixture (SGM) processing was integrated with a convolutional neural network (CNN) for binary and multi-class classification. They claimed that the SGM model increases detection and reduces the time cost. The proposed method was evaluated with 5 classes imbalanced technique and 2 classification algorithms. They were verified using the University of New South Wales-NB 2015 (UNSW-NB15) and the Canadian institute for cybersecurity intrusion detection system 2017 (CICIDS2017) datasets. The evaluation of the CICIDS2017 dataset shows that the method achieves an excellent detection rate of 99.85% in the 15-class classification. However, the detection rates for web attack brute force are still less than 50%, lower than random oversampling (ROS) and SMOTE. As for the UNSW-NB15 dataset, the detection rates for binary and 10 classifications reach 99.74% and 96.54%.

Abdulhammed *et al.* [23] used various techniques, such as over-sampling, under-sampling, spread subsample, and class balancer, to solve imbalanced data problems for binary classes. Several classifiers, such as random forest (RF), DNN, voting, variational auto-encoder, are used in the evaluation. The experiments on the CIDDS-001 dataset showed that DNN with the down-sampling method and class balancer is the most effective. By the experimental results, the class distribution has a light impact on the classification process. Furthermore, Abdulhammed *et al.* [36] proposed the uniform distribution based balancing (UDBB) for imbalanced classes. To reduce features, the auto-encoder (AE) and principle component analysis (PCA) were used in evaluation with various classifier methods. The simulation results on the original distribution of the CICIDS2017 dataset showed that PCA produces better accuracy than AE, at 99.6%. However, by implementing UDBB, the detection accuracy was reduced to 98.9%, although it better detected some attacks. In another experiment, Hua [29] used under-sampling and feature selection in pre-processing. The proposed traffic classification using LightGBM, based on the CSE-CIC-IDS2018 dataset. The model used only 10 features that were selected using Random Forest. They compared their models with various machine learning algorithms, and CNN deep learning. The best results for the overall accuracy obtained reached 98.37%. However, the influence of the model on the minority class was not discussed.

Yang *et al.* [37] applied an improved conditional variational autoencoder with a deep neural network in NIDS. An improved conditional variational AutoEncoder (ICVAE) training explores the relationship between data features and attack classes. This aims at balancing training data sets and improve detection performance in minority attack. They used cross-entropy as the function of reconstruction loss of the decoder. The results of this challenge showed that the best individual detection system obtains up to 89.08% and 85.97% of the multi-class classification in the UNSWB15 and NSL-KDD datasets, respectively. They claimed that ICVAE-DNN increases detection rates of minority and unknown attacks. Also, an unsupervised auto-encoder was used by Li *et al*. [38] to overcome imbalance problems in NIDS. They used the random forest to select significant features in the CSE-CIC-IDS2018 dataset, and performed anomaly detection for each attack. However, the results of AE-IDS for attacks, in web attacks (SQL injection, brute force web, and brute force-XSS), are still low and optimized. Similarly, the unsupervised auto-encoder model was used by Zhao *et al.* [39] They introduced the semi-supervised discriminant auto-encoder (SSDA) to overcome new attacks. Inspired by existing research, this study uses DAE to extract attack data. Furthermore, the focal loss is used to increase the detection rate of minority attacks. The CSE-CIC-IDS2018 dataset is used to test the model in multi-class classification and compare the impact of the three-loss functions on unbalanced processes.

3. RESEARCH METHOD

This study improves intrusion detection systems' ability to detect minority attacks class using deep learning models with deep auto-encoder (DAE) pre-tuning processes and fine-tuning using DNN. The classification process used 3 scenarios, including categorical cross-entropy (CE) loss, focal-loss (FL), and weighted categorical cross-entropy (WCE), as illustrated in Figure 1. The model was evaluated using CSE-CIC-IDS2018, which represents a recent attack dataset [40].

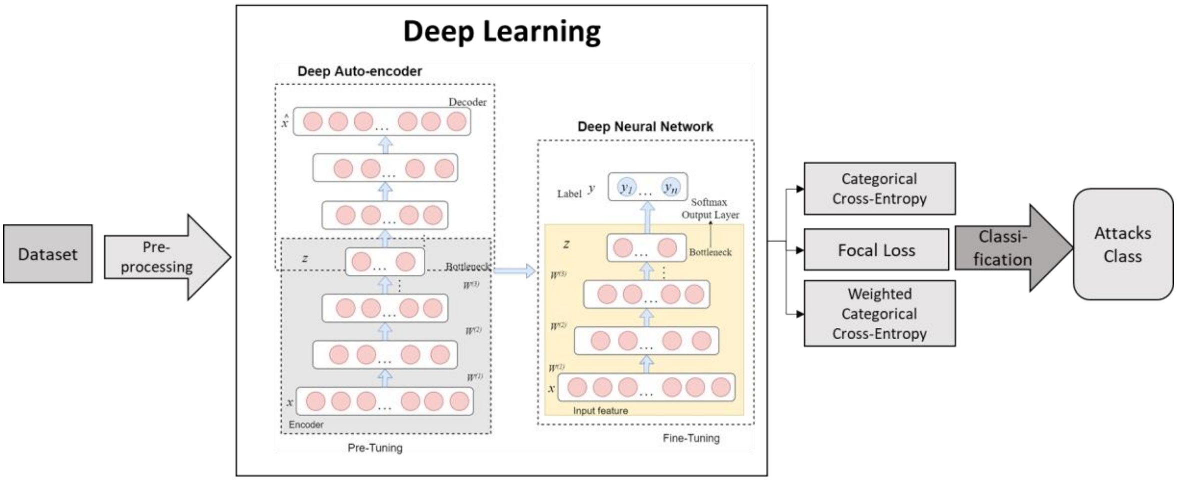


Figure 1. Deep learning architecture-based attacks classification

3.1. Dataset

The CSE-CIC-IDS2018 dataset consists of 80 features, including labels. The features of the dataset are generated and extracted with CICFlowMeter [40], [41]. The designed scenario consists of 6 attacks, including denial of services (DoS), distributed denial of services (DDoS), botnet, brute-force, web attacks, and infiltration, as presented in Table 1. They are grouped into 14 attack sub-classes. The total data amount is 16,232,943 records, dominated by 83.07% benign traffic. This study used 10% data for training and 2.5% for testing. About 51% of the data was used for benign composition. The structure of malicious data is used based on the amount of data. Table 1 clarifies the composition of the training and testing data used. The infiltration attack includes a stealthy attack that utilizes an internal network for illegal access. The characteristics of infiltration traffic and benign are very close, which implies a difficulty in detecting the network IDS [42]. As a result, the infiltration attack was eliminated in the experiment because this study discussed the focus, emphasizing the consideration of imbalanced class factors in accuracy detection.

Table 1. Composition of the CSE-CIC-IDS2018 dataset used

Category	Size		Train		Test	
Benign	13,484,708	83.07%	803,025	51.12%	201,238	51.21%
Bot	286,191	1.76%	85,842	5.46%	21,479	5.47%
BruteForce	380,949	2.35%	114,387	7.28%	28,465	7.24%
DDoS	654,300	4.03%	354,907	22.59%	88,593	22.55%
DoS	1,263,933	7.79%	212,053	13.50%	52,995	13.49%
Web Attacks	928	0.01%	754	0.05%	174	0.04%
Infilteration	161,934	1.00%	-	-	-	-
Total	16,232,943	100.00%	1,570,968	100.00%	392,944	100.00%

3.2. Pre-processing

From 80 features of the CSE-CIC-IDS2018 dataset [40], the timestamp feature was eliminated, and only 79 were used. The timestamp is encoded information that explains the occurrence of an attack. It is quite essential for prediction in time series. However, it is not essential for classification where the model must recognize the attack based on its characteristics. Feature flow duration has more impact on identifying attacks, such as DDoS and DoS, due to its rapid nature. In the detection and classification model of attack, the time of occurrence is not necessary. This is because, in its implementation, the attack happens at any time. Therefore, the feature is eliminated as in previous studies [26], [27], [38].

The first stage of dataset pre-processing is feature encoding, which transforms the data from categorical into numerical. The feature encoding process, using one-hot encoding, changed the protocol and label features into numeric data. The process mapped the protocol feature to 3 instances, including transmission control protocol (TCP), user datagram protocol (UDP), and Hop-by-Hop IPv6 (HOPOPT). Also, the label features became 6 feature attack categories by eliminating the infiltration class. Finally, 80 features for data and 6 feature labels were obtained. The next stage was the feature scaling process to turn all data values into the specified range. This process is necessary for features with a high value and not dominating others. Feature scaling uses the same approach of Min-max scaling with range [0, 1] as in a previous study [43]. After pre-processing, the data is ready for the training and testing process.

3.3. Deep learning architecture

The proposed model of the intrusion detection system is designed using the pre-tuning and fine-tuning process. The deep learning architecture used automatic feature extraction with deep auto-encoder (DAE) in the pre-tuning stage and the deep neural network (DNN) architecture in the fine-tuning stage. DAE performs the process of feature extraction with the encoding and decoding phase. Auto-encoder generates output $\hat{x}$, which is reconstructed from input x. In single-layer Auto-encoder, when the input vector $x \in \mathbb{R}$, the vector encoding function h, in forward propagation for hidden layer l $(l = 1)$, is notated as shown in (1). The decoding function is notated as shown in (2);

$$h = E(x) = f(W^{(1)}x + b^{(1)}) \tag{1}$$

$$\hat{x} = D(h) = g(W^{(2)}h + b^{(2)}) \tag{2}$$

W are weight matrices, b are bias vectors, x is an input vector form dataset, and *f(.) and g(.)* are activation functions used on a hidden layer. The experiment used several variants of ReLU activation functions, such as SeLU, PReLU, ELU, and Leaky ReLU in the hidden layer and sigmoid in the last layer of auto-encoder.

Deep auto-encoder (DAE) is the development of a single AE with a higher number of layers. The function compositions in the encoder and decoder are E and D, respectively the proposed architecture uses 7 hidden layers, with the output reconstruction $\hat{x} = D1(D2(D3(E3(E2(E1(x))))))$. For the reconstruction process input x into the output, $\widehat{x}$ uses the MSE loss function with (3). The backpropagation process produces a loss value close to zero

$$J(w, b, x^i, \widehat{x^\iota}) = \frac{1}{2}\left\|x^i - \widehat{x^\iota}\right\|^2 \tag{3}$$

A bottleneck layer (middle layer) with a smaller dimension of the input dataset represents the extracted features $z = E3 = f(W^{(3)}.E2 + b^{(3)})$. The result of the feature extraction in the form of an encoding structure and the weight and bias values are transferred to the deep neural network (DNN) structure for the fine-tuning training process. The encoding result of this structure becomes an input vector in the DNN classifier architecture. The DNN model uses the weight, bias and z value of the AE pre-training, to produce the output class prediction $\hat{y}$, written as;

$$\hat{y} = s(W^{(l)}.z^{(l)} + b^{(l)}) \tag{4}$$

Output $\hat{y}$ (prediction target label) will close to y (vector target label) by using the activation function softmax $s(.)$ in the last layer. For all the training datasets (x^i, y^i), the function of loss can be solved by:

$$J(W,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y^\iota}, y^i) = \frac{1}{m}\sum_{i=1}^{m} J(W,b;\ x^i, y^i) \tag{5}$$

where $\mathcal{L}$ is the loss function, and m is the number of datasets. One focus of this study is to compare the $\mathcal{L}$ loss function using cross-entropy, weighted cross-entropy, and focal loss.

The DNN model carried out the training process for classifying multi-class attacks. A hyperparameter tuning process is performed to get the best deep learning model by looking at the rate detection result of the attack classification. This tuning process tried various model variants based on the number of hidden layers, the number of nodes, learning rate value, batch-size, activation function, and kernel initialization to get the best model.

3.4. Loss function

In the case of a multi-class with the number of classes $(C > 2)$, the equation of the loss function for the categorical cross-entropy (CE) is:

$$\mathcal{L}(\hat{y}^i, y^i) = CE = -\sum_{i=1}^{c} y^i \log(\hat{y}^i) \tag{6}$$

C is the number of classes, y^i is the ground truth class, and $\hat{y}^i \in [0,1]$ is the model's predicted probability for the class. Where $y^i = 1$ belongs to the actual label of i; otherwise, it equals 0.

For imbalanced class cases, the CE loss function is modified by adding a weighting factor [44] to obtain the CE as shown in (7).

$$\text{Weighted } CE = -\sum_{i=1}^{c} \propto^i y^i \log(\hat{y}^i) \tag{7}$$

where α^i is the weight factor for class i.

The deficiency of CE loss is that many samples contribute to a significant accumulation of the loss value above the rare class [33], [34]. Therefore, when the extreme balance issue in the case of multi-class attack classification is resolved, the scenario takes advantage of the focal loss function proposed by Lin *et al.* [33]. The focal loss function does not provide the same weighted value on all training data. In contrast, focal loss reduces the weight of well-classified data. Its impact on focal loss emphasizes training on data that is difficult to classify with as shown in (8):

$$\mathcal{L}_{FC}(\hat{y}^i, y^i) = -\sum_{i=1}^{c} \alpha(1 - \hat{y}^i)^{\gamma}.y^i \log(\hat{y}^i) \tag{8}$$

with γ as a modularity factor to reduce the weight of well-classified classes. When $\gamma = 0$, the loss equals to cross-entropy. Therefore, $\gamma >= 0$ is set to evaluate the effect of samples classified with a loss factor. The parameter α is the weight to balance focal loss, and it increases the accuracy value for the imbalance class.

3.5. Experimental setup and performance metrics

The experiment was run on the cloud machine in the Google Colaboratory platform. The model was developed using the Python programming language with computation utilizing a TensorFlow-GPU library of Keras [45], a deep learning framework. The hyperparameter tuning process used Talos Library [46].

This observation used accuracy, sensitivity, and specificity to measure the performance of the proposed model. The evaluated performance used the accuracy function to assess the model's ability to classify attacks correctly. In the case of imbalanced datasets, the predicted result was dominated by large numbers of classes. Therefore, it is necessary to examine the model's specificity and sensitivity for imbalanced data set case [47]. The sensitivity results showed how precisely the model detected an attack. The specificity showed the probability that the model does not make mistakes in recognizing an attack.

4. RESULTS AND ANALYSIS

4.1. Hyper-parameter tuning

The hyper-parameter tuning process is crucial in obtaining a network architecture (number of neurons and layers). Moreover, the process was used to obtain the most appropriate hyper-parameter values in the deep learning model. In the initial phase, several hyper-parameter processes were performed on several hidden layers and nodes, batch size, learning rate, activation function, and kernel initial for deep learning model (DAE-DNN). The experiments used categorical cross-entropy as the lost function. The best architecture obtained for the CE loss on the CSE-CIC-IDS2018 dataset is the DAE structure, with 7 hidden layers (80-70-40-30-25-30-40-70-80). It was extracted and transferred to the model DNN to become 80-70-40-30-25-6. This architecture was obtained by trying different variations in the number of hidden layers DAE [1, 3, 5 and 7]. The best learning rate value used was 0.001, with the values of [0.00001, 0.0001, 0.001, 0.01, 0.1]. The experiment used the initial lecun_uniform kernel, as well as the Leaky ReLU activation function. The Leaky ReLU was previously selected through the tuning process of various activation functions. The batch size value for the best model used is 256. Also, this was obtained through the tuning process with batch size variations 32, 64, and 256.

After getting the best model with the CE loss function as the basis of comparison, tuning for the focal loss parameter was performed. Two parameters were tuned in a multi-class focal loss, in which the settings of γ reduced the effect of the modulation factor. The α parameter is the weight factor for the class. The focal loss parameters tuned are range γ ∈ [0, 5] and α ∈ [0, 1], as recommended in [33].

4.2. Result of various of focal loss

The effectiveness of the focal loss function in attack classification was measured by taking the best tuning result in focal loss parameters. The training process was performed with the number of epoch=30. Table 2 summarizes the overall hyper-parameter tuning results for the focal loss function, with various values of γ and α. Also, the weight assigned to the rare class has a stable range. However, it interacts with γ, making it necessary to select the two parameters together, as shown in Tables 2 (a) and 2 (b). In general, α increased slightly as γ fluctuated. In this case, α = 0.5 works best when γ = 1. The best results are the accuracy value of 98.223%, the sensitivity of 98.223%, and specificity of 99.814% for the entire attack classes.

The proposed model reached the highest accuracy metric at γ=1. It is reasonable because γ minimizes the loss contribution of the dominant class sample that is easily classified. When parameter γ increases, the probability of correct classification $(1 - \hat{y}^i)$ decreases. This probability increases the weight of minority class samples that are difficult to be classified. As a result, the model focuses on the difficulty class of classified samples that lowers classification accuracy.

Table 2. Experimental results with a variety of values α and γ for classifying attacks with 30 epoch training processes. (a) CSE-CIC-IS2018 with α-balanced CE achieves at most 98.21% accuracy. (b) In contrast, using FL with the same network with varying γ/α gives accuracy at 98.223% at γ=1 and α=0.5 settings

(a) Varying α for CE loss (γ=0)

A	Accuracy (%)	Sensitivity (%)	Specificity (%)
0.1	98.17	98.17	99.79
0.25	98.14	98.14	99.70
0.5	98.19	98.19	99.79
0.75	**98.21**	**98.21**	**99.79**
1	98.16	98.16	99.77

(b) Varying γ for FL (w. optimal α)

γ	α	Accuracy (%)	Sensitivity (%)	Specificity (%)
0	0.75	98.210	98.210	99.792
0.1	0.1	98.185	98.185	99.788
0.2	0.75	98.152	98.152	99.807
0.5	1	98.220	98.220	99.813
1	0.5	**98.223**	**98.223**	**99.814**
2	0.5	98.119	98.119	99.778
5	0.75	98.062	98.062	99.719

4.3. Performance and comparison

The results of configuring NIDS with focal loss (NIDS-FL) were evaluated by comparing them with cross-entropy loss (NIDS-CE), and weighted cross-entropy loss (NIDS-WCE) accordingly. Equal values of network architecture (number of hidden layers, number of nodes), and hyper-parameter value were used. The NIDS-CE and NIDS-WCE configuration do not use the γ and α. The weighted cross-entropy used a balanced mode. It means that this function replicates the smaller class until the number of samples in the minority and larger classes is equal.

In the first stage, training was conducted using epoch=30. After the training process, an evaluation was performed using testing data. Figures 2 show all the metric comparisons with various variants of the loss function. It shows that for epoch=30, almost the models' overall performance using the focal loss function was better than CE and WCE. Respectively, the accuracy value is 98.23%, precision to 98.34%, recall (sensitivity) to 98.23%, and specificity to 98.25%, as shown in Figure 2 (a). This research used a multi-class classification for BoT, Brute Force, DDOS, DoS, and web attacks. The results showed that NIDS's performance using focal loss was higher than cross-entropy and weighted cross-entropy, as presented in Figure 2 (a). The detailed result for the entire class may be observed in Table 3. The proposed model excellently detected BoT and DDoS attacks, with an overall performance above 99.9%. The overall performance for DoS attacks was higher than 90%, while for Brute Force, recall performance attained 94%, with a precision of only 84%. The file transfer protocol (FTP)-brute attack has a characteristic that resembles the slow-hypertext transfer protocol (HTTP)

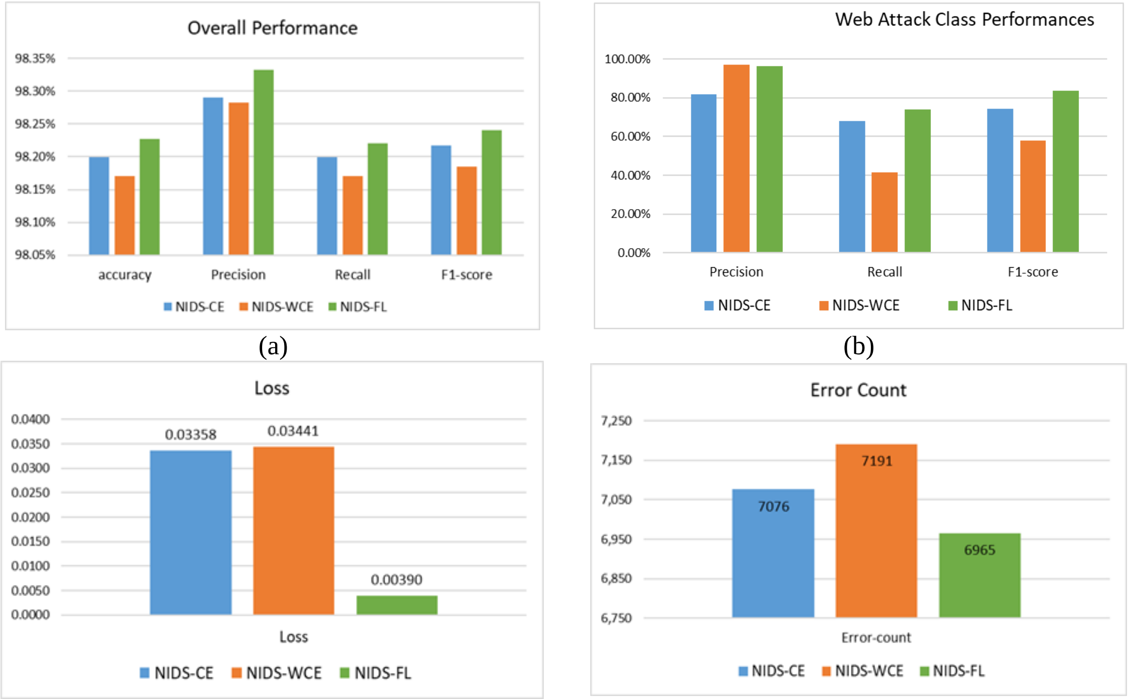
Dos. Both of them are often misclassified as a result, this reduced the recognition performance of both types of attacks.

Table 3. Performance of each attack class for epoch=30

	Benign	BoT	Brute Force	DDoS	DoS	WebAttack
			Precision (%)			
NIDS-CE	99.96	99.96	**84.2**	99.96	96.08	81.94
NIDS-WCE	99.94	99.98	83.77	99.82	**96.54**	**97.3**
NIDS-FL	**99.97**	**99.98**	83.91	**99.98**	96.5	96.27
			Recall (%)			
NIDS-CE	99.97	99.97	93.16	99.98	90.6	67.82
NIDS-WCE	99.92	99.95	**94.01**	99.98	90.21	41.38
NIDS-FL	**99.97**	**99.97**	93.97	**99.99**	**90.32**	**74.14**
			F1-Score (%)			
NIDS-CE	99.97	99.96	88.45	99.97	93.26	74.21
NIDS-WCE	99.93	99.96	88.6	99.9	93.27	58.07
NIDS-FL	**99.97**	**99.97**	**88.65**	**99.98**	**93.31**	**83.77**

To investigate the performance of the loss function against the imbalanced dataset, this study examined the model's efficiency in classifying types of attacks, especially in minority classes. The web attacks are a minority class that only amounts to 0.05% of the total data trained in Table 1. According to Table 3 and Figure 2 (b), the NIDS-FL outperforms the other methods to classify web attacks. There is a significant increase in the value of precision, recall, and f1-score compared to models that use CE and WCE losses. The recall (sensitivity) reaches 74.14%, implying an approximate increase of 7% from CE as a primary loss function. The model that uses WCE in minority classes with epoch=30 does not have excellent sensitivity, although it has a good precision value.

The loss value of the 3 models in Figures 2 (c) and, the FL function, is smaller than the CE and WCE. Also, the number of misclassification attacks were compared. The error count value of NIDS-FL in Figure 2 (d) is lower than other models, which are only 6965. The superiority of the FL shows the effect of modularity and weight factors on focal loss. By selecting the most appropriate modularity and weight for the imbalanced class, the loss and misclassification values can be minimized, especially for the minority class.



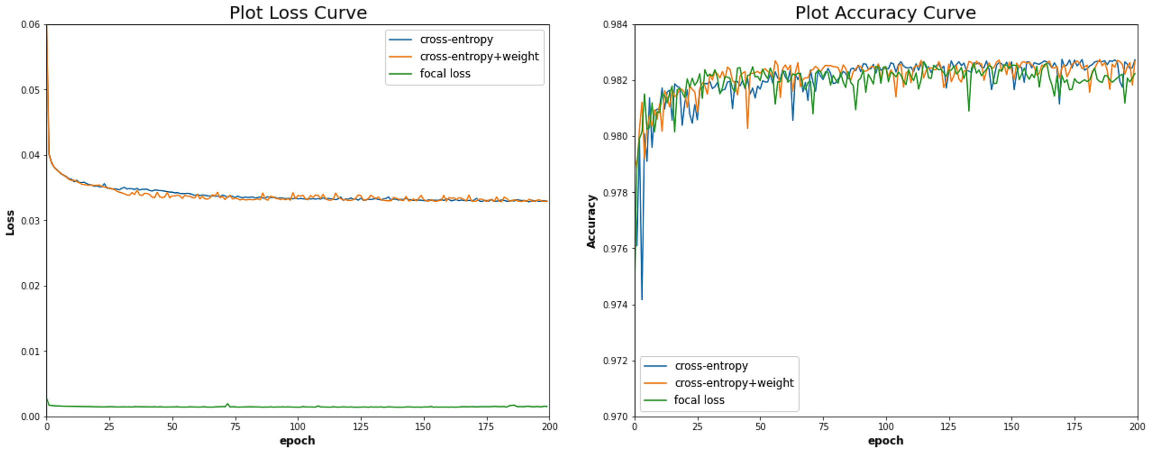
(a) (b)

(c) (d)

Figure 2. Graphs of comparison of various testing result metrics for all models (NIDS-CE, NIDS-WCE, and NIDS-FL) for the process epoch = 30; (a) Overall performance of attacks classification; (b) Performances of minority class-Web Attack class; (c) Loss value of all model; (d) Error count of all model

In the next phase, the model was evaluated by increasing the number of epochs to 200. As shown by the loss achieved in Figure 3 (a), the proposed deep learning model using FL with the previously selected hyper-parameters converge faster than CE and WCE. Figure 3 (b) shows the network using FL stabilizes after around 30 epochs, which is in contrast to 100 epochs and 80 epochs with CE loss and WCE loss, respectively. However, with the increasing number of epochs, the models using the cross-entropy function are more stable. Also, they tend to keep increasing compared to models that use focal loss and weighted cross-entropy functions. The improvement is reasonable because the hyper-parameter tuning process was performed on a model with a cross-entropy loss function. It has produced a model with the most appropriate hyper-parameters for deep learning. The resulting focal loss curve tends to fluctuate due to factor γ. Therefore, with an oscillating curve, the model has a slightly higher validation value when the curve reaches its peak.

Table 4 shows the best comparison of results for the 3 models after 200 epochs. The overall performance results are almost the same based on accuracy, precision, recall, F1-score, and specificity. For instance, the recall value indicates a tiny difference of <0.01%. In the web attack as a minority class, the overall performances after 200 epochs for models that use focal loss function for the precision, recall, accuracy, and F1-score values, respectively, amounted to 97.76%, 75.29%, and 85.07%.



(a) (b)

Figure 3. NIDS validation accuracy during training with CE loss, weighted WCE, vs. focal loss; (a) comparison of loss value; (b) comparison of accuracy

Table 4. Comparison of the Highest performances of the NIDS models presented in this study

	Training Set Performances in % (epoch=30)							Testing Set Performances in % (epoch=30)						
	Acc. (overall)	Prec.	Recall/ Sens.	F1-score	Specif.	Loss	Error Count	Acc. (overall)	Prec.	Recall/ Sens.	F1-score	Specif.	Loss	Error Count
NIDS-CE	98.20	98.29	98.20	98.22	99.80	0.03404	28330	98.20	98.29	98.20	98.22	99.80	0.03358	7076
NIDS-WCE	98.16	98.28	98.16	98.18	99.79	0.03472	28892	98.17	98.28	98.17	98.19	99.78	0.03441	7191
NIDS-FL	98.22	98.33	98.22	98.24	99.81	0.00400	27962	98.23	98.34	98.23	98.25	99.81	0.00390	6965
	Training Set Performances in % (epoch=200)							Testing Set Performances in % (epoch=200)						
NIDS-CE	98.26	98.38	98.26	98.28	99.82	0.03261	27303	98.27	98.38	98.27	98.29	99.81	0.03282	6814
NIDS-WCE	98.26	98.37	98.26	98.28	99.82	0.03284	27380	98.26	98.36	98.26	98.28	99.81	0.03272	6852
NIDS-FL	98.27	98.38	98.27	98.29	99.82	0.00140	27218	98.27	98.38	98.27	98.29	99.82	0.00138	6788

Models with deep auto-encoder pre-training process have advantages in overcoming the imbalance problem. An increase in the layers of deep auto-encoder network raises the number of complicated features learned from the original data. Transfer-layer with 4 deep encoding layers in DAE (7 hidden layers) improves the DNN fine-tuning process's classification results, despite extreme class imbalance problems. Out of the 3 models, although in the web-attack class where the training data only 0.05% of the total data, the sensitivity value reaches 74.14%, especially in those that use the focal loss function. However, in the web-attack class, the training data only 0.05% of the total data. In the model that uses cross-entropy, the results are acceptable, with a sensitivity reaching almost 68%. On the weighted cross-entropy model, the sensitivity is only 41.38%. The weights and bias in the DNN network are initialized according to transferred value from the encoding layer on DAE. The weighted cross-entropy, which uses the weight factor proportional to the class frequency, affects the training that is unsuitable for the positive samples at the beginning of the learning process.

The final loss layer of the deep neural network is the softmax loss function. In DAE optimization, the weights are initialized using lecun_uniform. As a result, the output from each layer is uniformly distributed. The value of the output layer in the deep neural network for softmax functions is uniformly distributed. Appropriately, the minor positive samples are less critical in the initial training stage. Therefore, in deep learning using focal loss, the last layer's bias term is initialized to some non-zero value [33]. More focus is directed at the positive examples in the early training stage, and the whole training process is likely to be effective. In deep learning, cross-entropy loss, and weighted cross-entropy loss was based on weight initialization. Different loss functions produced varied predictions on various models. In 30 epochs, the model used FL and achieved slightly better accuracy than CE loss. Another advantage of using the FL function is that the value of the cost loss is near zero. The loss of the cost function represents how well the model learns concerning the training examples.

In general, without modifying the distribution of the CSE-CIC-IDS2018 dataset, the model results were satisfactory. These results were better than the previous study using deep learning and resampling techniques in the same dataset, as shown in Table 5. The overall accuracy obtained at 98.27% was better than the deep learning model developed by Lin *et al*. [26]. An accuracy of 96.2% was achieved by Lin *et al*. [26], using SMOTE algorithm for imbalance class, and DNN model based on long short term memory (LSTM) based and, combined with attention mechanism (AM). Also, their study compared the various machine learning techniques with datasets that have been conducted over-sampling. All of the multiple methods performances are not significant compared to the proposed model in this study. For instance, for web attack classes as minority samples, they claimed the models developed reached 98% for a better recall value than the model in this study, which is only 75.29%. However, the precision value and F1-score for the web attack class is only 30%. In contrast, the model precision and F1-Score for web attack classes with focal loss in this study is 97.76% and 85.07%, respectively, after 200 epoch training.

This study is superior to the precision value and F1-score compared to the performances of previous research, as detailed in Table 5. Hua [29] used various machine and deep learning, as well as under-sampling and feature selection techniques for pre-processing. The recall value of the model with LightGBM was around 0.1%, slightly higher than the model proposed in this study. However, the precision value of this model is 0.24% superior to the model they proposed. Unfortunately, their research did not explain the effect of the model on web attacks as minority classes. Zhao *et al*., [39] with the semi-supervised discriminant auto-encoder (SSDA), and Ferrag *et al*., [48] with Deep Auto-encoder, utilized unsupervised learning without modifying the data distribution on the dataset. However, with the hyper-parameter process performed in this study, the deep learning model proposed resulted in better detection. This model works better with less data proportion and deep learning using focal loss. As a result, it solves much of the imbalance-class problem.

This study has successfully demonstrated the significance of deep learning. This has been achieved using deep auto-encoder as a feature reduction technique with focal loss functions. It has provided better results in terms of several performance metrics for IDS, especially in imbalance classes. However, there have been certain limitations and constraints in this study. In the evaluation process, the infiltration attack class in the initial test was eliminated without modifying the dataset. This is because it often caused misclassification with benign class. However, future studies should develop a deep learning model with a two-stage classification that detects infiltration attacks. Also, it is believed that the focal loss function optimizes the imbalanced class problem using other deep learning algorithms. Subsequently, future studies should try to evaluate the influence of the focal loss function with different deep learning algorithms. This study used the CSE-CIC-IDS2018 dataset in the training and testing processes. Future research should cover an anomaly-based online intrusion detection system.

Table 5. Comparison of the proposed model and previous studies in CSE-CIC-IDS2018 dataset

Reference	Handling ImbalancedClass Approach	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[26]	SMOTE	MLP	90	91	89	-
		LSTM	93	91	93	-
		LSTM+AM	96.2	96	96	-
[29]	Under-sampling	LightGBM	98.37	98.14	98.37	98.21
		MLP	97.58	97.23	97.58	97.37
		CNN	98.06	97.60	98.06	97.67
[39]	Original Distribution	SSDA	97.09	-	98.00	-
[48]	Original Distribution	CNN	97.38	-	97.28	-
		DA	97.37	-	98.18	-
Our Method	Original Distribution	DAE+DNN (focal loss)	**98.27**	**98.38**	**98.27**	**98.29**

5. CONCLUSION

This study presented a new deep learning model to address the problem of classifying multi-class attacks. The network architecture was partitioned into automatic feature extraction with deep autoencoder using 7 hidden layers, and a classifier with a fully connected deep neural network. The focal loss function was adjusted to the proposed model in an imbalanced dataset. The proposed deep learning model used the focal loss to obtain a faster convergence than cross-entropy loss and weighted loss. Concerning web attack classes as minority samples, the evaluation results of the CSE-CIC-IDS2018 show that the deep learning method with focal loss is a high-quality classifier with 98.38% precision, 98.27% sensitivity, and 99.82% specificity. Several future studies should be built on this research in several aspects. First, using the focal-loss function on imbalanced datasets should be evaluated by comparing them with various datasets. In this research, the infiltration attack class was eliminated, which behaves in the same way as benign traffic. However, future studies should improve a deep learning model that uses two stages to filter the infiltration attacks.

ACKNOWLEDGEMENTS

This research has received funding from Indonesia Ministry of Research, Technology, and Higher Education (grant agreement 170/SP2H/LT/DRPM/2020). The authors would like to thank data science research group, Universitas Bina Darma, and University for support and facilities.

REFERENCES

[1] F. Li, R. Xie, Z. Wang, L. Guo, J. Ye, P. Ma, and W. Song, "Online Distributed IoT Security Monitoring with Multidimensional Streaming Big Data," *IEEE Internet Things Journal*, vol. 7, no. 5, pp. 4387-4394, May 2020, doi: 10.1109/JIOT.2019.2962788.

[2] K. R. Sollins, "IoT Big Data Security and Privacy Versus Innovation," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1628-1635, Apr. 2019, doi: 10.1109/JIOT.2019.2898113.

[3] D. Stiawan, Mohd. Y. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, "Investigating Brute Force Attack Patterns in IoT Network," *Journal of Electrical and Computer Engineering*, vol. 2019, pp. 1–13, Apr. 2019, doi: 10.1155/2019/4568368.

[4] K. Yoshigoe, W. Dai, M. Abramson, and A. Jacobs, "Overcoming invasion of privacy in smart home environment with synthetic packet injection," *2015 TRON Symposium (TRONSHOW)*, Dec. 2015, pp. 1-7, doi: 10.1109/TRONSHOW.2014.7396875.

[5] B. Yang, L. Guo, F. Li, J. Ye, and W. Song, "Vulnerability assessments of electric drive systems due to sensor data integrity attacks," *IEEE Trans. Ind. Inf.*, vol. 16, no. 5, pp. 3301-3310, May 2020, doi: 10.1109/TII.2019.2948056.

[6] M. M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the trustworthiness in the industrial IoT networks through a reliable cyberattack detection model," *IEEE Trans. Ind. Inf.*, vol. 16, no. 9, pp. 6154-6162, Sep. 2020, doi: 10.1109/TII.2020.2970074.

[7] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, Jan. 2019, doi: 10.1109/COMST.2019.2896380.

[8] J. A. Jupin, T. Sutikno, M. A. Ismail, M. S. Mohamad, S. Kasim, and D. Stiawan, "Review of the machine learning methods in the classification of phishing attack," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 4, pp. 1545-1555, Dec. 2019, doi: 10.11591/eei.v8i4.1344.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525-41550, Apr. 2019, doi: 10.1109/ACCESS.2019.2895334.

[10] J. Majidpour and H. Hasanzadeh, "Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1137–1148, Jun. 2020, doi: 10.11591/eei.v9i3.1724.

[11] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, Dec. 2015, doi: 10.1186/s40537-014-0007-7.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.

[13] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, Jun. 2015, doi: 10.1007/s10489-014-0629-7.

[14] R. K. Meleppat, C. Shearwood, S. L. Keey, and M. V. Matham, "Quantitative optical coherence microscopy for the *in situ* investigation of the biofilm," *J. Biomed. Opt*, vol. 21, no. 12, p. 127002, Dec. 2016, doi: 10.1117/1.JBO.21.12.127002.

[15] K. M. Ratheesh, L. K. Seah, and V. M. Murukeshan, "Spectral phase-based automatic calibration scheme for swept source-based optical coherence tomography systems," *Phys. Med. Biol.*, vol. 61, no. 21, pp. 7652-7663, Nov. 2016, doi: 10.1088/0031-9155/61/21/7652.

[16] S. Nurmaini, *et al.*, "Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks," *Future Generation Computer Systems*, p. S0167739X20305410, Jul. 2020, doi: 10.1016/j.future.2020.07.021.

[17] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," *Internet of Things*, vol. 3-4, pp. 82–89, Oct. 2018, doi: 10.1016/j.iot.2018.09.003.

[18] Y. Zhang, P. Li, and X. Wang, "Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network," *IEEE Access*, pp. 1–1, Mar. 2019, doi: 10.1109/ACCESS.2019.2903723.

[19] M.-J. Kang and J.-W. Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," *PLOS ONE*, vol. 11, no. 6, p. e0155781, Jun. 2016, doi: 10.1371/journal.pone.0155781.

[20] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced Intrusion Detection System," Sep. 2016, pp. 1-8, doi: 10.1109/ETFA.2016.7733515.

[21] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35365-35381, May 2018, doi: 10.1109/ACCESS.2018.2836950.

[22] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 1–1, Jun. 2018, doi: 10.1109/COMST.2018.2847722.

[23] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic," *IEEE Sens. Lett.*, vol. 3, no. 1, pp. 1-4, Jan. 2019, doi: 10.1109/LSENS.2018.2879990.

[24] C. Xu, J. Shen, X. Du, and F. Zhang, "An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units," *IEEE Access*, vol. 6, pp. 48697-48707, Aug. 2018, doi: 10.1109/ACCESS.2018.2867564.

[25] M. Reza, S. Miri, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, 2016, doi: 10.14569/IJACSA.2016.070603.

[26] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic Network Anomaly Detection System by Using Deep Learning Techniques," *Cloud Computing-CLOUD 2019*, vol. 11513, pp. 161-176, Jun. 2019, doi: 10.1007/978-3-030-23502-4_12.

[27] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177, p. 107315, Aug. 2020, doi: 10.1016/j.comnet.2020.107315.

[28] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150-32162, Feb. 2020, doi: 10.1109/ACCESS.2020.2973219.

[29] Y. Hua, "An Efficient Traffic Classification Scheme Using Embedded Feature Selection and LightGBM," *2020 Information Communication Technologies Conference (ICTC)*, May 2020, pp. 125-130, doi: 10.1109/ICTC49638.2020.9123302.

[30] N. M. Sheykhkanloo and A. Hall, "Insider Threat Detection Using Supervised Machine Learning Algorithms on an Extremely Imbalanced Dataset," *International Journal of Cyber Warfare and Terrorism*, vol. 10, no. 2, pp. 1-26, Apr. 2020, doi: 10.4018/IJCWT.2020040101.

[31] W. Dai, K. Yoshigoe, and W. Parsley, "Improving Data Quality Through Deep Learning and Statistical Models," *Information Technology-New Generations*, vol. 558, pp. 515–522, 2018.

[32] W. Dai, I. Wardlaw, Y. Cui, K. Mehdi, Y. Li, and J. Long, "Data Profiling Technology of Data Governance Regarding Big Data: Review and Rethinking," *Information Technolog: New Generations*, vol. 448, pp. 439-450, Apr. 2016, doi: 10.1007/978-3-319-32467-8_39.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, Feb. 2018, doi: 10.1109/TPAMI.2018.2858826.

[34] G. S. Tran, T. P. Nghiem, V. T. Nguyen, C. M. Luong, and J.-C. Burie, "Improving Accuracy of Lung Nodule Classification Using Deep Learning with Focal Loss," *Journal of Healthcare Engineering*, vol. 2019, pp. 1–9, Feb. 2019, doi: 10.1155/2019/5156416.

[35] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the Robustness of Deep Neural Networks via Stability Training," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, Jun. 2016, pp. 4480–4488, doi: 10.1109/CVPR.2016.485.

[36] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection," *Electronics*, vol. 8, no. 3, p. 322, Mar. 2019, doi: 10.3390/electronics8030322.

[37] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network," *Sensors*, vol. 19, no. 11, p. 2528, Jun. 2019.

[38] X. Li, W. Chen, Q. Zhang, and L. Wu, "Building Auto-Encoder Intrusion Detection System based on random forest feature selection," *Computers & Security*, vol. 95, p. 101851, Aug. 2020, doi: 10.1016/j.cose.2020.101851.

[39] F. Zhao, H. Zhang, J. Peng, X. Zhuang, and S.-G. Na, "A semi-self-taught network intrusion detection system," *Neural Comput & Applic*, vol. 32, no. 1, Apr. 2020, doi: 10.1007/s00521-020-04914-7.

[40] C. I. for C. University of New Bruncswick, "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," 2018, [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018/ and https://www.unb.ca/cic/datasets/ids-2018.html

[41] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization" *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108-116, doi: 10.5220/0006639801080116.

[42] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, “Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks,” pp. 17, Dec. 2019, doi: 10.22667/JISIS.2019.11.30.001.

[43] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, “Automatic Features Extraction Using Autoencoder in Intrusion Detection System,” *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Oct. 2018, pp. 219-224, doi: 10.1109/ICECOS.2018.8605181.

[44] Y. S. Aurelio, G. M. de Almeida, C. L. de Castro, and A. P. Braga, “Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function,” *Neural Process Lett*, vol. 50, no. 1, pp. 1937–1949, Oct. 2019.

[45] W. Dai and D. Berleant, “Benchmarking Contemporary Deep Learning Hardware and Frameworks: A Survey of Qualitative Metrics,” *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, Dec. 2019, pp. 148-155, doi: 10.1109/CogMI48466.2019.00029.

[46] Talos, “Hyperparameter Experiments with Tensorflow, PyTorch and Keras,” [Online]. Available at: https://autonomio.github.io/talos/#/ (accessed Jun. 07, 2021).

[47] P. Banerjee, F. O. Dehnbostel, and R. Preissner, “Prediction is a balancing act: importance of sampling methods to balance sensitivity and specificity of predictive models based on imbalanced chemical data sets,” *Front. Chem.*, vol. 6, p. 362, Aug. 2018, doi: 10.3389/fchem.2018.00362.

[48] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2020, doi: 10.1016/j.jisa.2019.102419.

BIOGRAPHIES OF AUTHORS

Yesi Novaria Kunang is a doctoral student at the Doctoral Engineering Departement, Faculty of Engineering, Universitas Sriwijaya (Indonesia). She is a research scientist and a lecture in Faculty of Computer Science in Universitas Bina Darma. She is joined on Intelligent Systems Research Group (ISRG) Universiti Sriwijaya and Communication Network and Information Security (COMNETS) Universitas Sriwijaya. She obtained a Master's Degree in Computer Science from Universitas Gadjah Mada (Indonesia) in 2002. Her researches are in fields of security, intrusion detection, intelligent system, deep learning and also forensic computer. She is affiliated with IEEE as a student member. In the Journal of Computer Security, Matrik journals, and other scientific publications, she has served as invited reviewer. Further info on her homepage: https://www.researchgate.net/profile/Yesi Kunang.

Siti Nurmaini is a Professor in the Faculty of Computer Science University of Sriwijaya, Indonesia. She is a senior researcher and a founder of The Intelligent Systems Research Group (ISRG) Computer Engineering Department, Faculty of Computer Science, Universitas Sriwijaya. She obtained a Doctor's Degree in Computer Science from Universiti Teknologi Malaysia in 2011. Her work focused on the application of machine learning and deep learning in signal and image analysis techniques to medical diagnosis, especially in cancer, cardiac disease, and Covid 19. She is affiliated with IEEE and has served as an invited reviewer in MDPI journal, ComengApp journal and other scientific publications Further info on her homepage: https://www.researchgate.net/profile/Siti Nurmaini

Deris Stiawan (SCOPUS ID: 36449642900). He is an Associate Professor in the Faculty of Computer Science University of Sriwijaya, Indonesia. He is a member of IEEE, and since 2010 he is joined on Pervasive Computing Research Group (PCRG) Universiti Teknologi Malaysia. His professional profile has derived to the computer and network security fields, focused on network attack and intrusion prevention/detection system. In 2011, He holds Certified Ethical Hacker (C—EH) & Certified Hacker Forensic Investigator (C-HFI) licensed from EC-Council USA and Cisco Certified Networking Associate since 2005. In Telkomnika journal, IJECE, IJ-CLOSER, IJ-ICT, JIT, and other scientific publications, he has served as an invited reviewer and editor. Further info on his homepage: http://deris.unsri.ac.id/

Bhakti Yudho Suprapto is a lecture in the Electrical department in the Faculty of Engineering University of Sriwijaya, Indonesia. He obtained a Doctor's Degree in Electrical Engineering from Universitas Indonesia in 2018. His professional profile has derived to Robotic and Control, focused on, fuzzy logic and neural network.