Publications

8-18-2021

Learning to Detect: A Data-driven Approach for Network Intrusion Detection

Zachary Tauscher
Embry-Riddle Aeronautical University, tauschez@my.erau.edu

Yushan Jiang
Embry-Riddle Aeronautical University, jiangy2@my.erau.edu

Kai Zhang
Embry-Riddle Aeronautical University, zhangk3@my.erau.edu

Jian Wang
Embry-Riddle Aeronautical University, WANGJ14@my.erau.edu

Houbing Song
Embry-Riddle Aeronautical University, h.song@ieee.org

Follow this and additional works at: https://commons.erau.edu/publication

Part of the Systems and Communications Commons

Scholarly Commons Citation

Tauscher, Z., Jiang, Y., Zhang, K., Wang, J., & Song, H. (2021). Learning to Detect: A Data-driven Approach for Network Intrusion Detection. *IEEE Internet of Things Journal,* (). Retrieved from https://commons.erau.edu/publication/1769

This Article is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Publications by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

Learning to Detect: A Data-driven Approach for Network Intrusion Detection

Zachary Tauscher, Yushan Jiang, Kai Zhang, Jian Wang, Houbing Song
Department of Electrical Engineering & Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, FL 32114 USA
{tauschez, jiangy2, zhangk3, wangj14}@my.erau.edu, h.song@ieee.org

***Abstract*—With massive data being generated daily and the ever-increasing interconnectivity of the world's Internet infrastructures, a machine learning based intrusion detection system (IDS) has become a vital component to protect our economic and national security. In this paper, we perform a comprehensive study on NSL-KDD, a network traffic dataset, by visualizing patterns and employing different learning-based models to detect cyber attacks. Unlike previous shallow learning and deep learning models that use the single learning model approach for intrusion detection, we adopt a hierarchy strategy, in which the intrusion and normal behavior are classified firstly, and then the specific types of attacks are classified. We demonstrate the advantage of the unsupervised representation learning model in binary intrusion detection tasks. Besides, we alleviate the data imbalance problem with SVM-SMOTE oversampling technique in 4-class classification and further demonstrate the effectiveness and the drawback of the oversampling mechanism with a deep neural network as a base model.**

***Index Terms*—Intrusion Detection System, Machine Learning, Data Analytics, Computer Networks, NSL-KDD**

I. Introduction

Global communication and networking are commonplace in the current era. Everything from cell phones to thermostats is connected to the internet. A large number of users and devices connected to the internet makes the security risk to these networks only that much greater. The ability to detect and prevent network attacks is vital to maintain the confidentiality, integrity, and availability of our information and communication systems. Network intrusion detection and prevention systems (IDS/IPS) are a critical part of any network or system architecture designed to record and analyze connection behavior to identify possible attacks and report such information to an administrator or prevent the attack entirely.

IDPS technologies vary in their methodologies for detecting intrusions but tend to fall into two specific categories, signature-based and anomaly-based detection. Signature-based IDS systems, also known as misuse IDS, have been the most widely used due to their simplicity and reliability. These types of systems utilize pattern recognition to compare signatures of well-known attacks to current connections [1]. Anomaly-based IDS technology analyzes normal network traffic to develop models of normal behavior. Any connections that then deviate from these models are flagged as an intrusion. Anomaly-based IDS often produce a high volume of false positives as any activity that deviates from the normal is flagged. So while signature-based IDS is more often used, anomaly-based IDS has greater potential power, especially as machine learning and AI models continue to develop and become a greater focus in cybersecurity [2].

With the development of capable AI-driven IDS/IPS technologies, there have been various studies investigating and developing data-driven methods. Besides the data analytics and pattern presentations using traditional visualization techniques [3] and unsupervised K-means clustering [4] , the method toward detecting attacks can be mainly divided into two parts, the classical machine learning classifiers and deep learning models. In classical learning methods, several classifiers are utilized and modified for binary and multi-class intrusion detection tasks [5]–[8], including basic tree methods, Multilayer Perceptron, and Support Vector Machine, Naive Bayes, Random Forest, and a sophisticated variant of boost-based classifiers. Besides, feature selection is leveraged to choose the informative subset of features to facilitate the performance of classifiers, which is based on different techniques including Flexible Neural Tree [9], visualization techniques of distribution histograms, scatter plots, and information gain [10]. In deep learning methods, besides the deep neural networks [11], Convolutional Neural Networks [12], Recurrent Neural Networks [13], and their integration [14] are applied to capture certain spatial characteristics and temporal dependencies in an individual or joint manner, for downstream intrusion classification task. Moreover, to tickle the data insufficient issues in some datasets, transfer learning [15] and Variational Autoencoder [16] are also considered in terms of representation transferring and learning, which generalizes the learning-based methods to a wider range of applications.

In this paper, we aim at developing a data-driven intrusion detection framework to analyze and classify the patterns of normal network status and various malicious attacks. To be specific, we perform exploration on the NSL-KDD dataset representing real-world network traffic, by visualizing and analyzing potential patterns so that preliminary decision making and manipulation can be taken. Moreover, we adopt a two-stage hierarchy strategy based on machine learning models for intrusion detection tasks, where the attacks with abnormal patterns are first detected from normal samples, then further

Copyright © 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works by sending a request to pubs-permissions@ieee.org.

TABLE I
Attack Category and Its Statistics in NSL-KDD dataset

Attack Category	**Description**	**Attack Type & Count**
Probe	A process of probing target computer network to find weaknesses in its defense	satan (3633), portsweep (2931), nmap (1493), jpsweep (3599)
R2L	Unauthorized access from a remote machine to a local machine	spy (2), phf (4), multihop (7), imap (11), guess_passwd (53), ftp_write (8), warezmaster (20), warezclient (890)
U2R	Unauthorized access to local superuser privileges by a local unprivileged user	rootkit (10), perl (3), loadmodule (9), buffer_overflow (30),
DoS	Oversaturating connection bandwidth or depleting the target's system resources	teardrop (892), smurf (2646), pod (201), neptune (41214), land (18), back (956)

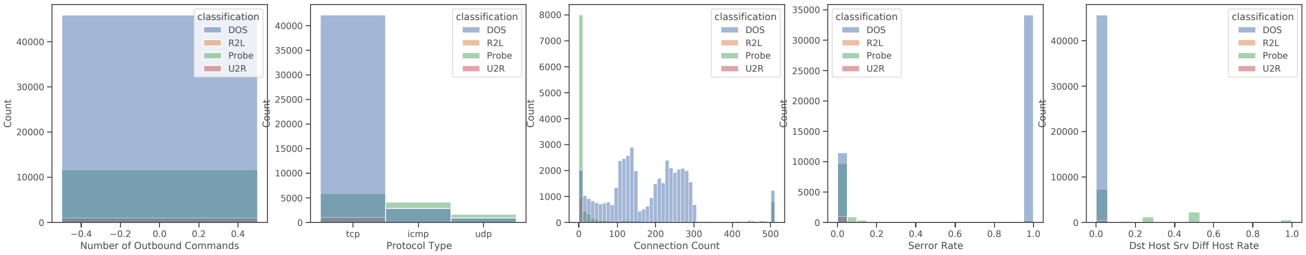

Fig. 1. Distribution histograms of specified features in the NSL-KDD data set.

classified into different types. At the first stage, we adopt supervised classifiers and a representation learning model to detect anomalies. As the intrusion data suffers from a severe imbalance problem, we first leverage an oversampling technique at the second stage, then utilize a deep neural network to classify the attack type in a supervised manner.

II. Data Exploration

In this section, we present the exploration of the NSL-KDD dataset, which includes the description, pattern visualization, and analytic.

A. Dataset Description

NSL-KDD is the refined version of the KDD'99 [17] data set to solve its inherent problems. For example, it does not contain redundant records so that the model training and evaluation is not biased by high-frequently duplicated records. Moreover, the number of selected records from each difficulty level group is inversely proportional to the percentage of records in the KDD'99 data set, which results in varying classification rates of different machine learning and it facilitates the analysis of distinct learning techniques.

1) Labels: The label of each instance in the NSL-KDD is assigned as either normal or an attack, with exactly one specific attack type. The attacks fall in one of following categories in Table I with the statistics summary of specific attack types, where *R2L* represents the *Remote-To-Local Attack*, *U2R* represents the *User-To-Root Attack*, and *DoS* represents the *Denial-of-Service Attack*.

2) Features: There are 41 features in the NSL-KDD data set that describe the characteristics of the cyber network, which can be further divided into three groups, consisting of 9 basic features, 19 traffic features, and 13 content features. The basic features involve all attributes that can be captured from a TCP/IP connection; the traffic features contain ”same host” and ”same service” features based on a connection window of 100 connections; the content features are related to suspicious behavior in the data portion like the number of failed login attempts. Generally speaking, traffic features are useful patterns to identify the DoS and probing attacks since they need to scan the hosts or send packets (many connections to some hosts) within a very short period of time. On the contrary, R2L and U2R attacks do not have any intrusion-frequent sequential patterns but are embedded in the data portions of the packets in a single connection. Hence, the content features are better patterns that can be used to detect these two attacks. A detailed explanation of each attribute is described in [18].

B. Data Visualization

To further understand and explore the NSL-KDD data set we used visualization techniques. Data visualization is the practice of graphically representing data. Using such methods we can gain insight and make better sense of large data sets By visualizing the NSL-KDD data set we gain a greater understanding of the features with relation to each other and attack type classification. For this investigation, we visualized the training data set to split up by attack type.

Our First steps in visualization are distribution histograms of the features in the data set, shown in Fig. 1. Distribution histograms plot the value of a feature against its occurrence in the data. From these graphs, we gain valuable information on issues within the data set, redundant features, and how features relate to different attack types. One of the first things we noticed during our initial investigation was the lack of instances of U2R and R2L attack types within distribution graphs. This is due to the small number of examples of these attacks within the data set. Further, we were able to discover some redundant features within the data set. Features 20 and 21 always have a value of zero, feature 20, the number of outbound commands. With a closer investigation of individual features, we can gain some insight into how specific features correlate to specific attack types. Fig. 1 shows some examples as to how these features correlate to specific attack types. We can see that most attacks use TCP. TCP has many vulnerabilities often exploited by attackers. DOS attacks often take advantage of the TCP handshake protocol by flooding a target host with incomplete connection and service requests in the hopes to waste server resources. As the server or host is attempting to handle a large number of connections from the attacker, it is not able to handle the requests of legitimate users thus rending the host inaccessible. This is reflected in the distribution graphs of connection count, which shows a large number of connections, and Serror Rate, which shows if those connection attempts had no further replies. Another example shown in Fig. 1 is with the Dst Host Srv Diff Host Rate histogram, which shows a correlation to the probe attack type. This graph shows the percent of connections to different destination machines from the same port number. Probe attacks will provide information on what each port is doing and what is using that port by sending information and waiting for a response. This nature is reflected in this graph as Probe attacks show up in greater numbers as this feature increases.

We further investigated the data set by calculating the correlation coefficient of the features compared to each other which can be seen in Fig. 2. This provides us insight into the strength of the relationship between the two figures. The greater the correlation the closer the value is to -1, or 1. Fig. 2 shows a strong correlation between higher-level figures which shows that these higher-level figures have a higher potential to provide information. These higher-level figures are often based on each other which can also explain why they have such high significance. Examples of these relationships are shown in the scatter plots of Fig. 3. Here we can see correlation between higher level features which provide information on he probe attack type. We can see that the probe attack type often threw the REJ flag, but this percentage was often affected by the number of services the probe reached out to. The further a probe attacked reached the greater its connection attempt was rejected. By observing these relations more directly we can determine the importance of some feature pairings towards anomaly detection and attack type classification.

III. Methodology

In this section, we present the pipeline of our detection mechanism. Firstly, we present the preprocessing details and

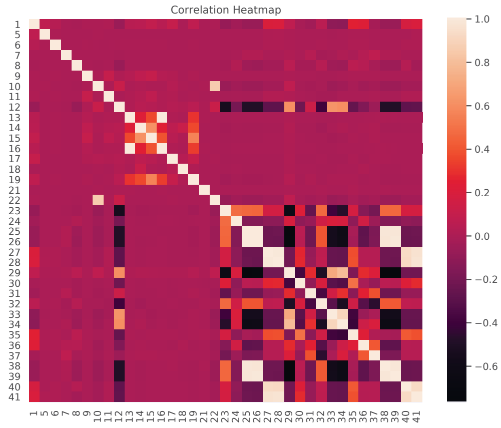

Fig. 2. Correlation heat map of features in the NSL-KDD data set.

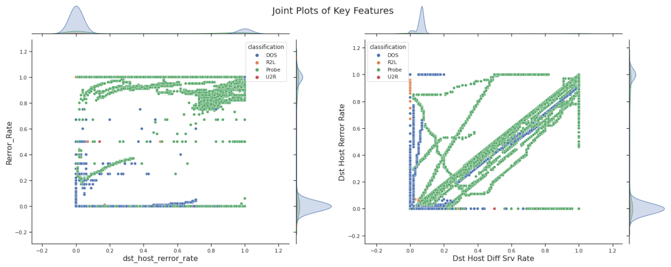

Fig. 3. Scatter plots of specific features in the NSL-KDD data set.

imbalanced nature of the explored dataset. After that, we also present a detection strategy and briefly describe the utilized machine learning algorithms. We adopt a hierarchy two-stage detection method: a binary classification followed by a 4-class classification. In the first stage, each input is classified into two normal samples and anomalies. We will explore different learning algorithms, including supervised learning classifiers and an unsupervised learning model, autoencoder. In the second stage, anomalies are further classified into four main attack categories (DoS, Probe, R2L, U2R), where supervised learning models are leveraged.

A. Preprocessing

Although the NSL-KDD data set is a cleansed data set, we still need preliminary preprocessing feature engineering before the data is fed into the model. To be specific, the categorical features should be converted into the numerical form so that they can be thought of as a vector in the Euclidean space: Three attributes (*'protocol_type','service',and 'flag'*) are categorical, we encode them by using a LabelCount encoder which sorts the categories by the frequency of each category within the feature. LabelCount has specific advantages at the outlier-insensitive nature and a reduction of dimensionality when certain features have very large numbers of categories.

After assigning numerical values to each categorical feature, the next step is to normalize each feature, as features that are measured at different scales do not contribute equally to the analysis and can create a bias for models. Therefore, the standardization as shown in Equation 1 is applied to transform the data to comparable scales (around the center 0 with a standard deviation of 1).

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where Z denotes the standardized feature, x denotes each value within the feature, μ, σ denotes the mean and standard deviation of all values, respectively.

In this dataset, the number of examples across the classes for the binary classification (normal vs. others) is roughly close. However, the 4-class intrusion data suffers from a severe imbalance, as the ratio of each class is approximately $920 : 220 : 20 : 1$. Most machine learning algorithms assume or expect a balanced class distribution for pattern learning and downstream classification tasks. When such data skewness exists, these algorithms fail to properly represent the distributive characteristics of the data whose results provide invalid accuracy across the classes of the data [19]. To alleviate the negative effect of imbalanced data set, we employ SVM-SMOTE [20], a sophisticated oversampling technique leveraging a Support Vector Machine algorithm to detect sample to use for generating new synthetic samples under the framework of Synthetic Minority Oversampling Technique (SMOTE) class [21].

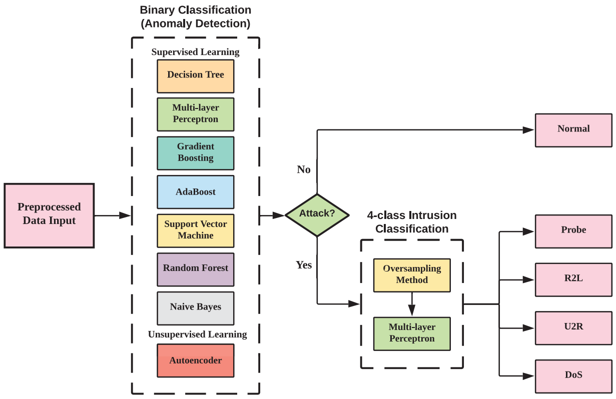

Fig. 4. Learning-based on hierarchy intrusion detection strategy.

B. Learning to Detect Network Intrusion

In this paper, we adopt various learning models for binary and 4-class intrusion detection. Fig. 4 depicts the overview of our IDS based on a hierarchy machine learning strategy. For binary detection, we utilize supervised learning models including Decision Tree, Random Forest, Naive Bayes, Support Vector Machine (SVM), AdaBoost, Gradient Boosting, Multi-layer Perceptron (MLP). Besides, we also consider an unsupervised representation learning model, Autoencoder, and treat the conventional binary classification problem as an anomaly detection problem. It learns the representation of normal samples and uses the reconstruction error as the anomaly score. At the testing stage, samples with high reconstruction (exceeding the threshold) are considered anomalies, as it is assumed that anomalies are difficult to be reconstructed [22]. The specific algorithm of Autoencoder-based detection is shown in Algorithm 1. After anomalies are detected, we utilize a deep neural network accompanied with the aforementioned oversampling technique to classify the specific intrusion type in a more robust manner.

Algorithm 1 Autoencoder-based attack detection algorithm

```
Parameters: Normal dataset $\mathbf{X}$, Anomalous dataset $x^i$, $i = 1, ...N$, threshold $\alpha$ defined by validation loss
Parameters: $f_\theta$ : Encoder, $g_\phi$ : Decoder
Output: reconstruction errors, anomaly indicator
$g_\phi, f_\theta \leftarrow$ train a Autoencoder with normal dataset $\mathbf{X}$.
LOOP Process
for $i = 1$ to $N$ do
    reconstruction error $(i) = \left\| x^{(i)} - g_\theta\left(f_\phi\left(x^{(i)}\right)\right)\right\|^2$
    if reconstruction error $> \alpha$ then
        $x^i$ is an anomaly (attack)
    else if reconstruction error $<= \alpha$ then
        $x^i$ is not an anomaly (attack)
    end if
end for
```

IV. Experiments

In this section, we evaluate the performance of different machine learning algorithms for two hierarchic stages of our intrusion detection system. We also explore the effectiveness of SVM-SMOTE oversampling technique toward a more valid classification model.

A. Evaluation Metrics and Experiment Settings

In this paper, we adopt accuracy, precision, recall, and F1 score for a comprehensive evaluation in the binary classification task. For 4-class classification, we use accuracy, F1 score of each intrusion type (one vs. all), with their macro-average (arithmetic mean) and micro-average (weighted mean) to demonstrate the classification performance of our model with the existence of imbalance [23].

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \ \times \ \text{Recall}}{\text{Precision} \ + \ \text{Recall}} \tag{5}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Next, we present the settings of the experiment for Autoencoder in binary classification and deep neural network in multi-class classification. The Autoencoder has three layers

TABLE II
PERFORMANCE EVALUATION OF BINARY CLASSIFICATION IN IDS

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	68.28%	68.16%	83.09%	0.7489
Random Forest	76.00%	87.34%	67.65%	0.7624
Naive Bayes	76.86%	96.21%	59.95%	0.7387
SVM	80.47%	**97.56%**	67.38%	0.7971
AdaBoost	79.40%	86.90%	75.14%	0.8059
Gradient Boosting	68.12%	65.04%	**95.13%**	0.7726
MLP	77.90%	95.82%	63.96%	0.7671
Autoencoder	**87.52%**	93.20%	84.22%	**0.8848**

with 15 neurons in hidden space and 0.15 Gaussian Noise, and 0.05 Dropout rate in encoding layers for regularization. The activation for both encoder and decoder is Scaled Exponential Linear Unit (SeLU), and the loss function is Mean Squared Error (MSE). In a deep neural network, three layers are used, where there are 80 neurons in the hidden layer with Rectified Linear Unit (ReLU) as activation function, four neurons in the output layer with Softmax as activation function. The loss function is cross-entropy. For both tasks, the batch size is 32; 0.15 of training data are used for validation; the optimizer is Adam with a learning rate 0.001; early stopping is adopted with the patience of 6 steps.

B. Binary Classification

The evaluation of different models for binary classification is shown in Table II, where the best result for each metric is indicated in bold while the corresponding second-best result is underlined. Among these learning models, it can be observed that SVM yields the highest precision score and the second highest accuracy, while Gradient Boosting classifier demonstrates its advantage on the highest recall score with a clear margin to the second highest one. In terms of the F1 score, AdaBoost and SVM achieve similarly good performance among supervised learning models. The above results illustrate that SVM and boosting methods gain more favors at separating attacks from normal samples within the scope of supervised learning.

On the other hand, the Autoencoder shows a huge advantage in the binary classification task, as it yields the highest accuracy, F1 score, and the second-highest recall score and. Moreover, the increase of accuracy and F1 score from the second-best result is 7.05% and 0.0789, respectively, demonstrating the power of unsupervised representation learning. To evaluate the classification performance with more details, the confusion matrix of Autoencoder is also presented, as shown in Fig.5. It is clear that Autoencoder performs slightly better on identifying normal behaviors than malicious attacks.

In general, the implemented Autoencoder with Gaussian noise and dropout as the regularization method is a good alternative in binary classification and anomaly detection. It is able to extract the feature information and generate salient and generalized vector representations for the reconstruction of normal samples only, where the anomalies usually do not share the similar representation space and fail to be reconstructed with a much higher loss during the model inference.

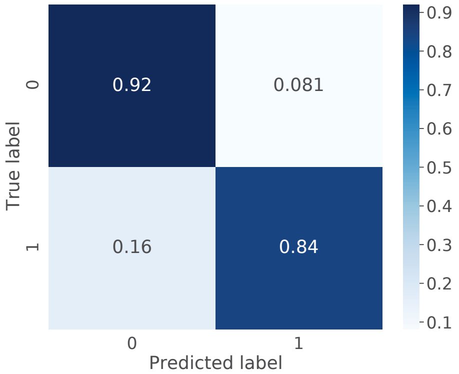

Fig. 5. Confusion matrix of Autoencoder based binary classifier.

C. 4-Class Intrusion Type Classification

At this stage, the samples identified as intrusion are further classified into different types. Table III shows the performance evaluation using a deep neural network for 4-class classification, without or with the aforementioned SVM-SMOTE oversampling method. It can be observed that the oversampling method has a minor impact on accuracy and micro F1 score, with the value around 80.4% and 0.783, respectively. However, it is clear that the new synthetic samples generated by SVM-SMOTE help the model to learn patterns and significantly improves the F1 score of U2R and yields an increase of macro F1 score by 0.1, with a slightly better F1 score of R2L and little inferior scores for DoS and Probe.

From Fig. 6, the confusion matrix of a 4-class classifier, we can conclude that despite the models are trained using the balanced data oversampled by SVM-SMOTE, it can only provide relatively accurate prediction on DoS (label 0) and Probe attack (label 1). In addition, a large portion of misclassified samples in R2L (label 2) and U2R (label 3) fall in the DoS attack, which suggests the inferior of our model at identifying the different patterns between DoS and R2L/U2R. Besides, a certain amount of misclassified samples in U2R fall in R2L, indicating a similar problem. One of the possible explanations is that, even with SVM-SMOTE to alleviate the imbalance problem for R2L and U2R, the patterns of these new synthetic samples appear to be insufficient to represent the attack behaviors in the testing set.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a data-driven intrusion detection framework based on data analytics and hierarchical learning-based detecting strategies. Firstly, we visualize potential patterns and discuss the relationships between features and underlying attack behaviors based on domain knowledge. Then, we leverage the supervised and unsupervised learning method to classify normal network behaviors and malicious attacks, and demonstrate the advantage of the unsupervised representation learning model in our task. Next, we focus on

TABLE III
PERFORMANCE EVALUATION OF 4-CLASS CLASSIFICATION IN IDS

Evaluation Metrics	**w/o Oversampling**	**w/ Oversampling**
Accuracy	80.47%	80.36%
$\mathrm{F1}_{DoS}$	0.8790	0.8678
$\mathrm{F1}_{Probe}$	0.8703	0.8560
$\mathrm{F1}_{R2L}$	0.4789	0.5100
$\mathrm{F1}_{U2R}$	0.2075	0.5741
Macro F1	0.6089	0.7020
Micro F1	0.7839	0.7836

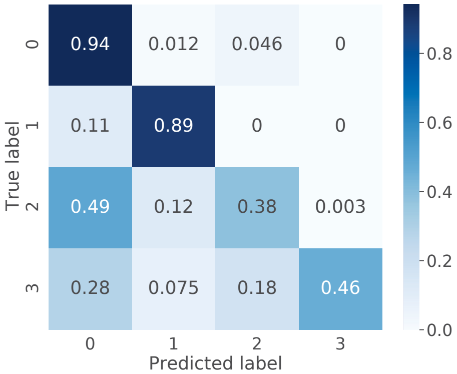

Fig. 6. Confusion matrix of deep neural network model for 4-class classification.

the problem of imbalance and alleviate it with SVM-SMOTE oversampling technique. We further demonstrate the effectiveness and the drawback of the oversampling mechanism and in 4-class classification with a deep neural network as a base model. In general, our framework yields satisfactory results on classifying normal samples and samples of the DoS and Probe attacks. However, it still shows certain inferiority in terms of minority class with an oversampling technique. For future work, we consider using more sophisticated learning models and techniques in terms of data imbalance, for example, cost-sensitive learning.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grant No. 1956193.

REFERENCES

[1] H. Holm, "Signature based intrusion detection for zero-day attacks: (not) a closed chapter?" in *2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 4895–4904.

[2] V. Jyothsna and V. V. R. Prasad, "Article: A review of anomaly based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, August 2011, full text available.

[3] M. J. Turcotte, A. D. Kent, and C. Hash, "Unified host and network data set," in *Data Science for Cyber-Security*. World Scientific, 2019, pp. 1–22.

[4] W. Zong, Y.-W. Chow, and W. Susilo, "Dimensionality reduction and visualization of network intrusion detection data," in *Australasian Conference on Information Security and Privacy*. Springer, 2019, pp. 441–455.

[5] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[6] P. Negandhi, Y. Trivedi, and R. Mangrulkar, "Intrusion detection system using random forest on the nsl-kdd dataset," in *Emerging Research in Computing, Information, Communication and Applications*. Springer, 2019, pp. 519–531.

[7] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: Kdd cup 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. IEEE, 2018, pp. 1–8.

[8] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 66–82, 2013.

[9] Y. Chen, A. Abraham, and J. Yang, "Feature selection and intrusion detection using hybrid flexible neural tree," in *International Symposium on Neural Networks*. Springer, 2005, pp. 439–444.

[10] R. C. Staudemeyer and C. W. Omlin, "Extracting salient features for network intrusion detection using machine learning methods," *South African computer journal*, vol. 52, no. 1, pp. 82–96, 2014.

[11] P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou, and D. Tzovaras, "An intrusion detection system for multi-class classification based on deep neural networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 1253–1258.

[12] P. Wu and H. Guo, "Lunet: A deep neural network for network intrusion detection," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 617–624.

[13] S. A. Althubiti, E. M. Jones, and K. Roy, "Lstm for anomaly-based network intrusion detection," in *2018 28th International telecommunication networks and applications conference (ITNAC)*. IEEE, 2018, pp. 1–3.

[14] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "Dl-ids: extracting features using cnn-lstm hybrid network for intrusion detection system," *Security and Communication Networks*, vol. 2020, 2020.

[15] P. Wu, H. Guo, and R. Buckland, "A transfer learning approach for network intrusion detection," in *2019 IEEE 4th international conference on big data analytics (ICBDA)*. IEEE, 2019, pp. 281–285.

[16] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42 169–42 184, 2020.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, 2009, pp. 1–6.

[18] L. Dhanabal and S. Shantharajah, "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[19] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[20] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[22] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[23] C. Liu, W. Wang, M. Wang, F. Lv, and M. Konan, "An efficient instance selection algorithm to reconstruct training set for support vector machine," *Knowledge-Based Systems*, vol. 116, pp. 58–73, 2017.