EKE, H., PETROVSKI, A. and AHRIZ, H. 2020. Handling minority class problem in threats detection based on heterogeneous ensemble learning approach. *International journal of systems and software security and protection* [online], 11(2), pages 13-37. Available from: https://doi.org/10.4018/IJSSSP.2020070102

Handling minority class problem in threats detection based on heterogeneous ensemble learning approach.

EKE, H., PETROVSKI, A. and AHRIZ, H.

2020

© 2020 IGI Global. All rights, including translation into other languages reserved by the publisher. No part of this journal may be reproduced or used in any form or by any means without written permission from the publisher, except for noncommercial, educational use including classroom teaching purposes.

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

SEE TERMS OF USE IN BOX ABOVE

DISTRIBUTED UNDER LICENCE

www.igi-global.com
International Journal of Systems and Software Security and Protection Volume 11 • Issue 2 • March, 2020

Handling Minority Class Problem in Threats Detection based on Heterogeneous Ensemble Learning Approach.

Hope Eke
Robert Gordon University, Scotland, UK
https://orcid.org/0000-0001-5049-8212

Andrei Petrovski
Robert Gordon University, Scotland, UK
https://orcid.org/0000-0002-0987-2791

Hatem Ahriz
Robert Gordon University, Scotland, UK
https://orcid.org/0000-0002-1389-3886

ABSTRACT

Multiclass problem, such as detecting multi-steps behaviour of Advanced Persistent Threats (APTs) have been a major global challenge, due to their capability to navigates around defenses and to evade detection for a prolonged period of time. Targeted APT attacks present an increasing concern for both cyber security and business continuity. *Detecting the rare attack is a classification problem with data imbalance. This paper explores the applications of data resampling techniques, together with heterogeneous ensemble approach for dealing with data imbalance caused by unevenly distributed data elements among classes with our focus on capturing the rare attack. It has been shown that the suggested algorithms provide not only detection capability, but can also classify malicious data traffic corresponding to rare APT attacks.*

Keywords: Imbalance data, Resampling techniques, Multi-steps, Multiclass classification, Oversampling, SMOTE, Recurrent Neural Network, Long Short-Term Memory, Gated Recurrent Unit and Ensemble techniques.

1. INTRODUCTION

The ability of an intrusion detection system to detect every possibility of an active attack on a system is a global security challenge. There have been a number of successful breaches of critical infrastructure. Stuxnet is one example of a sophisticated APT attack purposefully launched to target critical nuclear infrastructure in Iran as highlighted in (McAfee Labs, 2011) and (Chen et al., 2011).

There are diverse views as to what makes a threat an APT. Some believe that an APT is nation-state sponsored attack (Ahmad, Webb, Desouza, & Boorman, 2019), as a term that is frequently been used in security threat discussions (Smiraus & Jasek 2011), while (Five, 2011) and (ISACA, 2014) retain their definition, "APT is often aimed at the theft of intellectual property or espionage as opposed to achieving immediate financial gain and are prolonged, stealthy attacks". However, (Cressey, 2012), (Micro, 2013) and (Chen et al., 2018) view APT as a highly sophisticated combination of different techniques to achieve a specifically targeted and highly valuable goal.

This type of attack has drawn special attention to the possibilities of APT attacks on the Industrial Control System (ICS) such as Supervisory Control and Data Acquisition (SCADA) network. It has also led to research in developing methods to detect intrusions within a network and isolated devices at any level. Due to the dynamic and diverse nature of techniques used by attackers to implement an APT attack, these yielded to uneven distribution different classes. Hence, learning from imbalanced data has notable challenges for machine learning algorithms, since they need to deal with uneven distribution among examples of different classes in the training set (Krawczyk, 2016a) and (Zhou & Liu 2005). Handling imbalanced data distribution in multi classification problem based on ensemble supervised learning and problem decomposition with cost-sensitive learning are still an active research area in machine learning community as demonstrated by the authors (Nguyen et al., 2019), (Nguyen et al., 2018) and (Krawczyk, 2016b).

However, most of these proposed works has led to a significant pool of solutions geared towards addressing both binary and multiclass imbalance problem (Weiss, 2004). Majority of this solutions where mainly for binary imbalanced problem (Krawczyk, 2016a); hence, there is every need for research direction towards developing reliable solutions to deal with multiclass scenario problem. This paper focuses on the implementation of diverse data resampling techniques in combination with heterogeneous ensemble learning approach for handling multiclass imbalanced datasets with special interest on capturing the minority class.

The contribution of this paper can be summarised as follows:

- Implementation of several oversampling and undersampling approaches for handling binary and multiclass imbalance datasets with main focus on minority class in multiclass label.
- Analysing the impact of these approaches that could be used to improve the results obtained, without proposing a new algorithm or technique for handling imbalance data.
- Implementation of oversampling techniques for the multi-class imbalanced classification on two datasets (KDDCup99[1] and UNSW-NB15[2]), with close attention on the impact and knowledge of the minority class and imbalance distribution factors.
- Carried out series of experiments to: evaluate the impact of resampling imbalance data and ensemble deep neural networks to (i) accurately detect and classify an attack as abnormal and (ii) classify multiclass label into different type of attacks family.

The remainder of this paper is organised as follows. In the background section 2, we covered classification of imbalance data, techniques for learning from imbalance data and application in security domain together with an overview of deep neural network and ensemble learning approach as applicable to this study are discussed.

Experimental data, evaluation metrics and analysis results are discussed in Section 3. The conclusion of this study and future work are presented in section 4 and 5 respectively.

2. BACKGROUND

This section contains a brief background of imbalance data and applicable technique for handling imbalance distribution with respect to security domain based on literature review of few approaches and basic definition. This section also contains an overview of deep neural network and ensemble learning approaches as implemented on this study are discussed.

2.1 Learning from Multiclass Imbalance Data

Threats to network security remain one of the biggest challenges facing organisations and industries at various levels of operation. One of the major forms of such threats is referred to as APTs, which exploit multiple ways to open the system to malware. Classification and prediction has become an important task for pattern recognition as these events are rarely observed in any given network, making the classification and prediction task suffers from imbalanced data as stated by authors (Haixiang et al., 2017). This problem of imbalance data is not limited to security domain, but also applicable in detection of fraudulent transaction or call (Dal Pozzolo et al., 2014), text classification (Cardie & Howe, 1997) and several other domains. Hence the need for a classification model that will yield a high detection rate including identifying the rare class of interest.

The uneven distribution among examples of different attacks categories in classification tasks is known as the problem of imbalanced data whereby instances of certain classes occur more frequently than others (the minority class). This identified problem makes it difficult for learning algorithms as mentioned in (Sun et.al, 2009). Several studies have demonstrated that uneven data distribution is not the only factor that affects performance of diverse modelling classifiers as discussed by arthurs (Sáez et.al, 2016), (Batista et.al, 2004) and (Sun et al., 2009). Other factors may include the number of minority class, this could result to not enough data for training the model, minority class forming small distributed groups which leads to class separability problem, this is the main problem with minority class as highlighted by (Sun et al., 2009). Also, a high classification error may as well contribute to poor performance of the validation matrix implemented in any a given problem (Sáez et al., 2016). Again, class overlap as pointed out by (Vuttipittayamongkol et.al, 2018) is also known to have a higher impact on the classification of imbalanced datasets than the dominance of the majority class, although their proposed approach "new undersampling method that eliminates negative instances from the overlapping region" (Vuttipittayamongkol et al. 2018) has only been tested on binary class.

Diverse techniques have been proposed and successfully applied in handling imbalance data distribution problem, among which are but not limited to (i) data preprocessing, such as oversampling and undersampling techniques, decomposition of original problem into binary problem (Hoens et.al, 2012), however, implementing multiclass problem as binary may lead to loss of important information, (ii) cost-sensitive learning (Zhou & Liu 2005), implementing MetaCost (Domingos 1999), one-versus- one approach (OVO) and one-versus-all approach (OVA) based on decomposition schemes (Haixiang, et al., 2017) and implementation of ensemble modelling (Nguyen et.al, 2018) and (Nguyen et.al, 2019).

As an example, let us consider the multi-steps APTs detection problem, in which the percentage of the different dynamically generated transactions steps of APTs scenario comparing to the legitimate transaction of 99.98% is very low, as limited to 0.02 %. As the case of KDDCup99 dataset used in this study that contains four attacks categories with records of denial-of-service (DOS) as 391458, surveillance (Probe) as 4107, remote-to-local (R2L) as 1126 and user-to-root (U2R) as 52 of which R2L and U2R in essence are very rare (see Figure 2. for visual representation). However, any approaches which do not take into

consideration the imbalance distribution of class elements, may lead to increase difficulty of the classification task as observed in the previous study (Eke et al., 2019).

During the training, it was observed that LSTM-RNN appear to be more suitable for classifying high-frequency attacks and also the low frequency attacks with lower confidence prediction of 62.50%, 56.20% and 37.50% for LSTM, GRU and RNN respectively on multi-class attack detection task, while achieving a very significant average accuracy of 99.99% for LSTM (Eke et.al, 2019), although accuracy is not a recommended performance matrix for such task.

APT attack has affected many organisations, some of the victims of this modern attack do not provide any details about the attacks against them. As far back as 1998 with the first public recorded targeted attack named Moonlight Maze, that targeted Pentagon, NASA, the US Energy Department research laboratories and private universities and successfully compromised Pentagon computer networks and access tens of thousands of files (Smiraus & Jasek 2011). When examining the APT and the new methods being used to breach today's security controls, it distilled down to a basic understanding that attackers, especially those who have significant financial motivation, have devised effective attack strategies cantered on penetrating some of the most commonly deployed security controls (largely signature-based antivirus and signature-based intrusion prevention), most often by using custom or dynamically generated malware for the initial breach and data-gathering phase.

The advanced and persistent are major features that make APT differ from basic and traditional attacks (TA). These characteristics made APTs ideal for campaigns against enterprises as the perfect tool that easily penetrates defences, avoided detection for lengthy periods of time and stole sensitive information. Hence, the need for further investigation into handling imbalance data distribution in view to finding a better approach that could be considered to improve the achieved results.

2.2 Imbalance Learning Techniques

Diverse algorithms and techniques have been proposed and implemented for handling imbalance data problem. The basic strategies for dealing with imbalanced learning as described in (Das et.al, 2018), (Haixiang et al., 2017) and (Sáez, et al., 2016) are discussed in this section.

2.2.1 *Data-level approach*

This level involves application of preprocessing techniques such as resampling method in other to rebalance sample data distribution point so as to improve the accuracy of the learning process. Resampling method is classified into three group, undersampling instances from the majority class or oversampling instances from the minority class such that at the end, the number of labelled instances from both classes become comparable and hybrid methods (combination of oversampling and undersampling approaches)

- *Resampling Methods*

  Resampling methods are statistical procedures that use value of information from the observed data and modifying the class distribution of the data to draw certain conclusions about the data as described in (Chernick, 2012), (López et al., 2012) and (Sinharay, 2010). This resampling technique can be achieved through means of undersampling / oversampling and nonparametric statistical methods. More information about nonparametric methods can be find in (Chernick, 2012) and (Sinharay, 2010).

  - Oversampling techniques - such as Random Oversampling (ROS) and Synthetic Minority Oversampling Technique (SMOTE), involves process of oversampling instances from the minority class thereby creating new synthetic instances of that minority class (Chawla et al., 2002), (Dendamrongvit, & Kubat 2009) and (Giraldo-Forero et al., 2013).

- Undersampling techniques – example Random Undersampling (RUS): - undersampling instances from the majority class by discards examples from the majority classes until the effect of imbalance is significantly mitigated (Chawla et al., 2002), (Dendamrongvit, & Kubat 2009) and (Giraldo-Forero et al., 2013).
- *Hybrid Methods* – this method involves combination of oversampling and underrsampling approach. Example of such approach is SMOTETomek. In terms of computational time, a situation where within a given dataset, there are hundreds of minority instances, underrsampling method appear to be a better option to an oversampling method, while in a case of fewer observation of minority instances, the oversampling method SMOTE appear to be a better option (Haixiang et al., 2017).

- *Feature Selection (FS) and Extraction (FE)* - FS can be divided into filters, wrappers, and embedded methods, this method of data preprocessing select a subset of k features from the existing feature that will enable a classifier to achieve optimal performance, where k is a user-defined. Examples of FE techniques are PCA (Principal Component Analysis), SBD (Singular Value Decomposition) and NMF (f Non-negative Matrix Factorization), more details can be found in (Yijing, Haixiang., Xiao, Yanan, & Jinling, 2016).

2.2.2 Algorithm-Level Approaches

These methods adopt the existing learning algorithms and make a modification so as to reduce bias towards the majority class in favour of the minority class. These approaches can be viewed as

- *Boundary Shifting Methods* – this method uses disparate costs with the aim at artificially moving the decision boundary towards the majority class (Haixiang et al., 2017).
- *Discriminative regression based supervised learning models* - this approach was proposed and implemented by author in (Peng et.al, 2017).
- *Cost-Sensitive Learning* - in these methods, the minority class is assigned a higher cost of miss-classification compared to the majority class (López et.al, 2012), (López et.al, 2013) and (López et.al, 2015).
- *Active Learning* – active learning forms a part of the semi- supervised machine learning paradigm where the learner is allowed to interact with the user (or some equivalent source of information) to obtain the desired outputs at new data points, under the assumption that labelling can be expensive for large unlabelled datasets (Attenberg, & Ertekin, 2013).

2.2.3 Hybrid Approach

These methods involve the combination of various preprocessing techniques and algorithm-based approach so as to improve the robustness of this approach with respect to the nature of data imbalance distributions of the minority classes. Few examples of this approach as adopted for imbalance data classification are as follows.

- *Sampling Based Approaches with Cost-Sensitive Learning* – this approached involves combination of presprocessing data resampling techniques either by underrsampling or oversampling the imbalance data before implementing a classifier with cost-sensitive turning. Some remarkable work was carried out by authors (Chawla et.al, 2002), combined oversampling and underrsampling with C4.5 and Naïve Bayes clarifier, (López et al., 2012), combined cost-sensitive learning classifiers such as C4.5 with oversampling and underrsampling while (Akbani et.al, 2004) combined SMOTE with cost-sensitive SVM.

- *Sampling and Data Balancing Approaches with Classifier Ensembles* – these approach uses diverse oversampling, underrsampling together with data ensemble classifier mostly bagging and boosting such as SMOTEBoost with some balancing techniques while dealing with imbalance data. The authors (Nguyen, et al., 2018) combined ensemble classifier through class prototypes.

2.3. Imbalance Learning Classification Algorithms

2.3.1 Ensemble learning approaches

Can be seen as combination of multiple classifiers within a classification learning system. It is believed that within a wider range of applications and under a diverse scenario, combining several base classifiers has shown to yields a better performance to those of single classifier (Polikar, 2006), (Zhang & Duin, 2011) and (Nguyen et al., 2019). These approaches have been successfully used to address problem of imbalance data among other applications are: the main purpose of implementing ensemble systems is to reduce the chances of choosing a single poor performance classifier, or to improve a single poor performance classifier by using an intelligently combined ensemble of classifiers (Polikar, 2006).

Few examples of different taxonomy of ensemble learning methods are discussed below.

- *Iterative / Incremental Based Ensemble Learning* – These approaches involves the ability of a classifier to adopt to learn from new introduced class instances without forgetting the previously acquired knowledge. Examples are: Boosting, Adaboost**.** The advantage of using Adaboost is that it updates its weight distribution based on the performance of hypothesis *ht* generated in the previous iteration. Any samples that fail to be assigned to the correct class are given higher weights, this forces the future classifier to focus on learning these failed classified samples (Polikar, 2006) and (Haixiang et al., 2017). Incremental approach is proposed in (Parikh et al., 2004).
- *Parallel Based Ensembles* – Examples of parallel based ensemble methods include resampling-based ensembles, bagging and feature selection-based ensembles. In these methods, each base classifier can be trained in parallel, hence an advantage as time saving and ease of development, thus for dealing with practical problems (Haixiang et al., 2017).
- *Data Fusion* – these methods can be viewed in form
    - *Homogeneous Ensemble Methods*, where the same learning algorithm uses different training sets obtained from diverse data source to generate a set of classifiers, the combined outputs of these classifiers are used to make an informed decision about the phenomenon generating underlying data distributions (Haixiang et al., 2017), (Nguyen et al., 2018) and ((Nguyen et al., 2019).
    - *Heterogeneous Ensemble Method*: in these methods diverse learning algorithms are learned on the same training set to generate different base classifiers. It mainly involves combination of different techniques on the meta-data, aimed at achieving a high accuracy and the desired data fusion (Polikar, 2006), (Polikar et.al, 2006), (Nguyen, et al., 2018) and (Nguyen et al., 2019). Proposed Learn ++ in (Parikh, et al., 2004) and implemented on (Polikar et al., 2006), where Lean++ was used to generate an ensemble of classifiers that are been trained on available data to achieve a significant and constant output.

2.3.2 Recurrent Neural Network

Recurrent neural network (RNN) is an effective class of artificial neural network (ANN) that is used when dealing with complex supervised and unsupervised tasks (KP, 2019) and (Muñoz et al., 2019). Recently, deep learning techniques have been applied in cyber security (Eke et al., 2019) and (McDermott et.al, 2019). RNN and its variants has the capability to detect the cyber attacks by learning the complex underlying structure and hierarchical feature representations from a huge set of networks traffic data. The authors (Eke et al., 2019), proposed a novel approach and implemented using deep neural networks for APT multi-step

detection which takes stacked LSTM- RNNs networks to automatically learn features from the raw data to capture the malicious patterns. Although this approach achieved a significant average accuracy of 99.99% for LSTM, GRU and RNN on differentiating attacks from normal instances but the model tends to be more suitable for classifying high-frequency (majority classes) attacks and also the low frequency attacks (minority classes) with lower confidence prediction of 62.50%, 56.20% and 37.50% for LSTM, GRU and RNN respectively on multi attack detection. Also (KP, 2019). Proposed and evaluated RNN model against classical support vector machine classifier (SVM) for cybersecurity in Android malware classification, incident detection, and fraud detection.

RNN emerged as a powerful approach for deep learning architecture generally applicable for time-series data modelling. Despite the RNN and its variant networks remarkable performance in long standing AI sequence data modelling tasks such as time-series analysis, speech recognition and machine translation (Eke et al., 2019) and (LeCun et.al, 2015), applying the same in cyber security task is in early stage of development (Vinayakumar et.al, 2017).

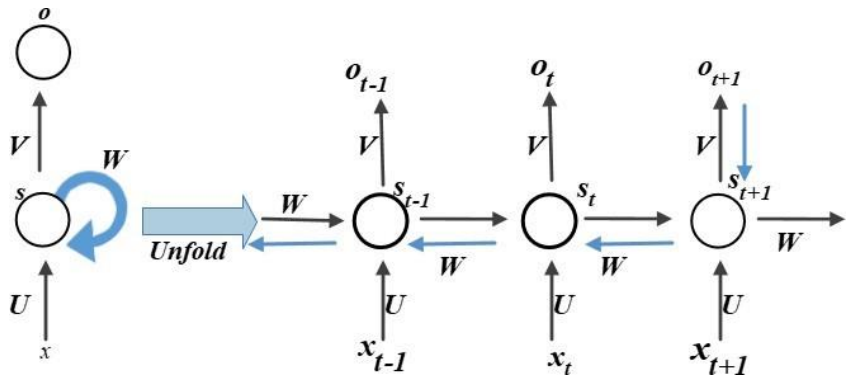


Figure 1. Schema of Unfolded Basic Recurrent Neural Network, modified after (Eke et al., 2019)

RNN were developed in the 1980s and it is similar to a feed- forward network (FFN) with an additional internal feedback loop (short-term memory to store and retrieve past information over time scales and thereby execute the temporal task) which is a circular connection between higher- and lower-layer neurons and optional self-feedback connections. These feedback connections enable RNNs to propagate data from earlier events to current processing steps which is in contrasts to FFN. The formulated mathematical representations of the RNN computational flow are represented in Equations 1 and 2

$$s_t = f_w(s_{t-1}, x_t) \qquad (1)$$

Equation 1 represent the basic recursive formula of RNN where f_w is the recursive function. The network has an input layer x, hidden layer s (also known as hidden state) and output layer o. The current observed input to the network at time step t is denoted as x_t , the hidden state at time step t is denoted as s_t and s_{t-1} represent the previous hidden state

$$s_t = f(Ux_t + Ws_{t-1}) \qquad (2)$$

s_t is calculated based on the previous hidden state s_{t-1} and the input at the current step x_t as shown in Equation 2. s_t is the short-term "memory" of the network which captures information about what happened in all the previous time steps, this can further be fed to other stacked recurrent layer or final

layer where the layer has nonlinear activation function such as softmax function *sf* as represented in Equation 3.

$$o_t = sf(Vs_t) \quad (3)$$

o_t is the output at time step *t* and the vector of probabilities of initial hidden state is set to *V*.

f denotes the nonlinearity mapping function from the input features to the output labels such as tanh or ReLU, where the weight matrices of the previous state weight at time step t and input state weight at time step t are represented as Ux_t and Ws_{t-1} respectively. s_{t-1} is required to calculate the first hidden state and is usually initialised to zeroes.

Understanding the dynamics of RNN entirely is difficult due to its cyclic connection. To overcome this an RNN structure input sequence of length is transformed to an FFN structure by unfolding over time-steps as represented in Figure 2. FFN consist of hidden layers. This new structure can be analysed and also is adaptable to the backward propagation (BP) of errors, at this point the predefined error function are computed by comparing the output values with correct values and then distributed back throughout the network layers. This process is often used to train deep neural networks (DNN). An unfolded RNN at any given time *t* is defined as a function h_T Equation 4.

$$h_T = S_T(x_2\, x_1) \quad (4)$$

Where S_T represents the unfolded graph of time-steps *t*.

The sum of all input-output pairs in a sequence over all the time- steps is referred to as the loss *L* function represented in Equation

$$L = d(tr, pr) = \sum_{t=1}^{T} d(tr, pr) \quad (5)$$

The schema of unfolding of RNN in time of the computation is shown in Figure 1. In RNN cyclic connections, each layer represents per time information. The unfolded RNN shares weight parameters *W* across time-steps as represented in Equation 2. This indicates the fact that network performs the same task with various inputs over time-steps. In addition to learning the temporal patterns with cyclic connections, unfolding allows the RNN model to learn the association of static features between the input and output sequences. In order to apply the feedback concept, the BP is used to compute the gradients for weight parameters across time-step *t*.

To find the recurrent weights, the computation of gradient at $t = 2$ will involve back propagating 1 step and add them to find and update the recurrent weight. This technique is known as back propagation through time (BPTT) employed by RNN to reduce network cumulative error. However, modelling large scale data sequence with RNN and BPTT is not efficient due to vanishing and exploding gradient problem as stated in (Bengio et.al, 1994) which usually occurs when we BP the error back in many time-steps in the deep unrolled RNNs network models.

Since RNN shares the parameters across all time step reducing the amount of parameter to be trained, this is utilised to calculate the gradient at each time *t* as in Equation 6 and

$$s_t = \tanh(W_{xs}\, x_t + Ws_{ss}s_{t-1} + b_s) \quad (6)$$

$$o_t = softmax(W_{so}\, s_t + b_o) \quad (7)$$

where s_t is the hidden layer, *tanh* is the hidden layer nonlinear activation function, the softmax function *sf* is used at the last layer as, b_s and b_O are the bias terms for the hidden state and prediction at time step t.

Where the prediction at time step *t* is denoted as o_s while the weight shared between the hidden layer s and output o across all the time sequence are W_{os} and W_{ss} respectively

2.3.3 Long Short-Term Memory (LSTM)

LSTM is a second order RNNs that is augmented by recurrent gates known as Forget Gates (FG) (Hochreiter & Schmidhuber, 1997). LSTM has the capability to remember information for long periods of time. It contains a memory block which is a complex processing unit that is composed of one or more memory cell and a pair of multiplicative gates known as input and output gate with in-built recurrent connection value 1 as constant error carousel (CEC). This value will be active across the time-step and triggered when a memory block has not received any value from outside signals (Ostaszewski et.al, 2006). In order to combat the issue of vanishing gradient that prevents RNN from learning long term dependencies through gating mechanism, the computation of recurrent hidden state *St* can be seen as mean elementwise multiplication as shown in Equation

$$\begin{aligned} i &= \sigma\left(x_t U^i + s_{t-1} W^i\right) \\ f &= \sigma\left(x_t U^f + s_{t-1} W^f\right) \\ o &= \sigma(x_t U^o + (s_{t-1} or) W^o) \\ g &= \tanh(x_t U^g + s_{t-1} or W^g) \\ c_t &= c_{t-1} + goi \\ s_t &= \tanh(c_t) oo \end{aligned} \qquad (8)$$

LSTM comprises of one cell state *c* and three gates. The input i forget f and output gates *o* used to illustrate interaction within LSTM architecture as represented within Equation 8. These gates are composed out of a sigmoid function σ that generate output vectors between 0 and 1 through elementwise multiplication operation *o* used in Equation 8 with another vector to decide how much of the newly computed state for the current input you want to let through by input gate i, the forget gate *f* defines how much of the previous state you want to let through. The output gate defines how much of the internal state to expose to next layer in time *t* step.

The candidate value ∂ is calculated based on the current input and the previous hidden state, the input gate i will decide which part of this information to store in cell state memory c_t as the new hidden state ∂. The internal memory of the unit is denoted as c_t, this is the combination of the previous memory c_{t-1} multiplied by the forget gate f and the newly computed hidden state g, multiplied y the input gate. Finally, the hidden state output state s_t can be computed by multiplying the internal memory c_t with output gate o.

2.3.4 Gated Recurrent Unit (GRU)

GRUs are gating mechanism in recurrent neural networks. Its performance on polyphonic music modelling and speech signal modelling are similar to that of LSTM with fewer parameters due to lack of output gate. A GRU has two gates, a reset gate *r*, and an update gate *z* shown in Equation 9

$$
\begin{aligned}
z &= \sigma(x_t U^z + s_{t-1} W^z) \\
kr &= \sigma(x_t U^r + s_{t-1} W^r) \\
h &= \tanh\left(x_t U^h + (s_{t-1} or) W^h\right) \\
s_t &= (1-z)oh + zos_{t-1}
\end{aligned}
\tag{9}
$$

3. EXPERIMENTAL STUDY

The purpose of this study is to examine the impact of implementing resampling method of handling imbalance data on multiclass classification in other to improve our previously implemented RNN variants algorithm in APTs detection (Eke et al. 2019). Two separate experiment were implemented. One experiment was implemented without application of resampling techniques and the second experiment involves implementation of SMOTE oversampling and NearMiss undersampling approaches on training dataset.

In this study, we have carried out three different tasks. The first task involves data preprocessing. All the standard data mining processes such as data cleaning and preprocessing, normalisation, visualisation and classification were implemented in Python. The batch size of 128 and epochs are run up to 50 and 100 with a learning rate set in the range of 0.01-0.5 on a GPU-enabled TensorFlow network architecture. RNN variants classification algorithms were used to perform the classification experiments on both dataset in order to analyse the network protocol relationship with the attack used by intruders in generating anomalous network traffic. The second task focused on deriving hyper-parameter values for best performance model and then applied the achieved hyper-parameter values in measuring the model performance. We also compared the result of this model to previously published work in (Eke et al., 2019) while the third task, involves implementing the two experiments as mention above.

The classification metrics result obtained without applying resampling techniques was compared to the result obtained when resampling techniques were applied in order to further evaluate the performance of RNN variants model on two different datasets (KDDCup99 and UNSW-NB15). This classification task analysis was implemented as a binary class and multiclass classification on both datasets used with main focus on minority class of interest within the multi-class classification. The training dataset were normalised from 0 to 1. This was trained using sigmoid activation function through time with ADAM optimiser, sigmoid function was used on all the three gates and categorical/binary cross entropy as loss function for multi and binary classification respectively.

3.1 Experimental Data

In this study, two different datasets (KDDCup99[1] and UNSW-NB15[2]) were used. The 10% KDDCup99 dataset containing 494021 records used consists of 22 attacks classes and 1 normal class. These 9 attacks were grouped into 4 main attack classes - Denial of Service (DOS), Remote to User (R2L), User to Root (U2R) and Probe. Figure 2 shows the number of records in each of the classes. All features were used as input vector with 70% as training set and 30% as testing set for the binary and multi classification respectively. The UNSW-NB15 dataset consists of 9 attacks classes and normal class. These 9 attacks were grouped into 6 main attack classes - Generic, Exploits, Fuzzers Denial of Service (DoS), Analysis, and

Worms. This UNSW-NB15 dataset has been partitioned into training and testing sets containing total records of 175331 training and 82332 testing datasets.

Figure 2 is a visual representation of classes in KDDCup99 and UNSW-NB15 datasets as used in this study, it shows a clear picture of class distribution in both datasets with A and B representing binary and multiclass data records in 10% KDDCup99 dataset, while C and D represent the 6 attacks class and normal class in UNSW-NB15 training and testing dataset.

A *10% KDDCup99 dataset in binary class*

B *10% KDDCup99 dataset in multiclass*

C *UNSW-NB15 training dataset*

D *UNSW-NB15 Testing dataset*

Figure 2. Visual representation of experimental datasets (10% KDDCup99 and UNSW-NB15 datasets)

3.2 Evaluation Metrics

Generally, accuracy is used as a traditional way of classification performance measure. This matric measure is no longer appropriate when dealing with multiclass imbalance data since the minority class has little or no contribution when compared to majority classes toward accuracy (Sun, et al, 2009). For this reason, this study considered precision, recall, f1-score, overall accuracy, area under the curve (AUC) receiver operating characteristic (ROC) and confusion matrix to get a clearer understanding of the output.

- *True Positive (TP)* - abnormal instances correctly predicted as abnormal.
- *True Negative (TN)* - normal instances correctly predicted as normal
- *False Positive (FP)* - normal instances incorrectly predicted as abnormal
- *False Negative (FN)* - abnormal instances incorrectly predicted as normal
- *Precision (P)* - is the ability of a classification model to identify only the relevant data points, that is the ratio of true positive (TP) records over the sum of true positive (TP) and false positive (FP). Represented as equation

$$P = \frac{TP}{TP+FP} \qquad (10)$$

- *Recall (R)* – which is also known as probability of detection, true positive rate (TPR) or sensitivity (S) is the ability of a model to find all the relevant cases within a dataset, that is the ratio of the true positive (TP) records over the sum of true positive (TP) and false negative (FN). Represented as equation

$$R = \frac{TP}{(TP + FN)} \quad (11)$$

- *F1-score (f1)* - is the weighted average of precision and recall, that is the harmonic mean of precision and recall of a class in one given metric, represented as in Equation 12, where weight parameter is denoted by β^2 generally set to 1 by default and which measures the trade-off between recall and precision. By default, β often set to

$$f1 = \frac{(1+\beta^2)\,.precision\,.recall}{\beta^2 precision + recall} \quad (12)$$

- *AUC-ROC curve* - The precision-recall curve shows the trade-off between precision and recall at different threshold settings, where AUC measures the degree of separability and ROC represent the probability curve and also a useful technique for model output visualising, organising and selecting classifiers based on their performance as described in (Fawcett, 2006). A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate. High scores for both precision and recall show that the model is returning accurate results (precision), as well as returning a majority of all positive results (recall). An ideal model with high precision and high recall will return many results, with all results labelled correctly. The AUC-ROC curve can also be used as a scaler measure rather than the higher the AUC value, the better the model (Haixiang et al., 2017).

$$\text{AUC} = \frac{1 + \text{TP}_{\text{rate}} - \text{FP}_{\text{rate}}}{2} \quad (13)$$

- Overall Accuracy (OaAcc) - Measures the rate of the correctly classified class instances of all the classes (attacks and normal). An overall classification performance is an important performance matrix require to evaluate the overall model performance rate, calculated as represented in Equation 14.

$$OaAcc = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (14)$$

3.3 Results and Discussions

To validate the impact of implementing resampling techniques and model performance on imbalance data for multi-class classification in detecting minority class of interest (U2R in KDDCup99 and Worms in UNSW-NB15 datasets) and multi-attacks, statistical matrices as recommended by (Haixiang et al., 2017), such as precision, recall, f1-score, AUC-ROC curve and overall average accuracy are calculated

- to evaluate the ability of the RNNs variants model (before and after resampling training data) to accurately detect and classify an attack as abnormal
- and also, to investigate how accurately this model will detect different type of attacks with our main focus on detecting the minority class of interest

Table 1 contains the comparative overall average summary result achieved by individual algorithms before and after applying SMOTE oversampling techniques on KDDCup99 and UNSW-NB15 datasets, while Table 2 and 3 shows the summary result of individual class classification performance and average mean classes detection rate of both resampled and non-resampled data for both datasets. The overall visualisation of these results as obtained in mult-class and binary class classification task are also represented in Figures 5(a & b) and 6(a & b) for KDDCup99 and UNSW-NB15 datasets respectively.

The RNNs variants and stacked-ensemble RNNs variants model were used as classifier and as a detector. As a binary classifier to separate normal from attacks instances. This model was able to achieve a noticeable overall average accuracy of probability confidence and macro f1 results of (99.95% & 99.93% and 99.95 % & 99.92% for KDDCup99) and (76.09% & 73.06% and 75.92% & 72.80% for UNSW-NB15) datasets on oversampled and non-oversampled data respectively, with insignificant performance output comparing result between oversampled and non- oversampled data. The individual algorithms performance can also be seen in Figure 6a and 6b for both datasets.

However, since the focus of this study is on detecting the minority class among the multiclass attacks, using the confusion matrix output in Figure 3, the chosen performance matric measures were calculated to get a better understanding of the overall model performance. A closer observation of these overall performance matric measures results as in Table 1 and the individual performance of each of these algorithms in classifying individual classes as represented in Table 2 and 3 shows a slightly different in result in detecting the minority class of interest in both resampled and non-resampled data for both datasets. As a detector, the overall average detection rate improved from 63.20%, 68.40% & 57.60% to 78.90%, 78.90% & 73.70% for minority class of interest (U2R) in KDDCup99 dataset (see Figure 5a), while in the case of detecting Worms in UNSW-NB15 dataset, although implementing SMOTE oversampling approach did make a huge impact as the overall average detection rate improved from 0.03%, 0.06% & 0.09% to 32.50%, 35.50% & 23.2% but the detection rate is still below average (see Figure 5b), this demonstrate that uneven data distribution is not the only factor that affects performance of diverse modelling classifiers as discussed also by arthurs in Sáez et.al, (2016), Batista et.al, (2004) and Sun et al. (2009). High classification error as highlighted in Sáez, et al., (2016) also contributed to the poor performance of the validation matrices implemented in this study.

For the case of KDDCup99 dataset, while there are some spikes in the validation accuracy and loss as shown in Figure 7 and 8, following the individual model accuracy and loss per epoch, achieving training and validation accuracy of 99.96%, 99.95 % (LSTM), 99.95%, 99.94% (RNN), and 99.96%, 99.95% (GRU) with overall validation average accuracy of 99.95% and average mean detection accuracy improving from 91.43% to 94.33% for oversampled and non-oversampled data. Also, the training and validation accuracy are quite close to each other, indicating that the model is not overfitting.

Furthermore, the AUC-ROC shows a significant improvement from 82% to 89%, 82% to 87% and 82% to 89% for U2R as can be visualised in Figures (9 to 11), with the rest of the classes achieving above 97% with micro and macro-average ROC curve of 100% and 97% respectively. While in the case of minority class of interest “Worms” in UNSW-NB15 dataset, AUC-ROC, improves from 50% to 59%, with micro and macro-average ROC curve of 73% and 65% respectively. With this macro-average obtained in both datasets, it indicates that the classifier performs well for each individual class.

Table 1. Overall summary performance result of class classification of both resampled and non-resampled data for 10% KDDCup99 and UNSW-NB15 datasets

Comparative overall performance result table for multiclass classification data analysis						
10% KDDCup99 dataset						
Criteria	LSTM		RNN		GRU	
No of Samples	494020 - 10% KDDCup99					
	SMOTE	No-Re	SMOTE	No-Re	SMOTE	No-Re
TP	148149	148149	148132	148126	148134	148147
FP	58	62	75	71	73	60
TN	148149	148149	148132	148126	148134	148147
FN	58	62	75	71	73	60
OaAc (%)	99.96	99.96	99.95	99.95	99.6	99.96
ValLoss (%)	0	0	0		0	0
	Scores					
AgAc (%)	99.98	99.98	99.98	99.98	99.98	99.98
P (%)	99.96	99.96	99.95	99.95	99.95	99.96
Se / TPR (%)	99.96	99.96	99.95	99.95	99.95	99.96
Sp (%)	99.96	99.96	99.95	99.95	99.95	99.96
f1 (%)	99.96	99.96	99.95	99.95	99.95	99.96
Voting by Probability Confidence						
	SMOTE	No-Re				
OaAgAc (%)	99.96	99.95				
macro f1 (%)	92.93	91.24				
UNSW-NB15 dataset						
Criteria	LSTM		RNN		GRU	
No of Samples	175331 Training and 82332 Testing - UNSW-NB15 dataset					
	SMOTE	No-Re	SMOTE	No-Re	SMOTE	No-Re
TP	36333	43559	41199	34403	30716	36333
FP	45999	38773	41133	47929	51616	45999
TN	36333	43559	41199	34403	30716	36333
FN	45936	38779	3997	47935	51049	45936
OaAc (%)	48.63	52.91	50.04	41.79	37.31	41.13
ValLoss (%)	2.75	2.23	2.52	2.35	3.44	3.98
	Scores					
AgAc (%)	77.73	81.02	78.95	74.71	69.45	76.14
P (%)	48.63	52.91	50.04	41.79	37.31	41.13
Se / TPR (%)	48.73	52.9	50.74	41.78	37.57	41.16
Specificity (%)	48.63	52.91	50.04	41.79	37.31	41.13
f1 (%)	48.68	52.9	50.39	41.78	37.44	41.15
Voting by Probability Confidence						
	SMOTE	No-Re				
OaAgAc (%)	48.37	48.11				
macro f1 (%)	30.1	31.1				

Table 2. Summary result of individual class classification rate and average mean classes detection rate of both resampled and non-resampled data for 10% KDDCup99 dataset

	Classes	Dos					Normal					Probe					R2L (class of interest)					U2R (class of interest)					AvgAc (%)
Algorithm	Metrics	De (%)	P (%)	Se(%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	
LSTM	SMOTE	100	100	100	99.99	100	99.97	99.9	99.99	99.98	99.91	99.99	99.68	99.51	100	99.59	99.98	96.79	93.26	99.99	94.99	99.99	57.69	78.95	99.99	66.67	99.98
	No Re	100	100	100	99.99	100	99.96	99.91	99.9	99.98	99.91	99.99	99.35	99.68	99.99	99.51	99.97	94.71	94.44	99.99	94.58	99.99	66.67	63.16	100	64.86	99.98
KNN	SMOTE	99.99	99.99	100	99.97	100	99.95	99.91	99.86	99.98	99.88	99.99	99.19	99.43	99.99	99.31	99.98	95.45	94.38	99.99	94.92	99.99	51.85	73.68	99.99	60.87	99.98
	No-Re	99.99	99.99	100	99.98	100	99.96	99.86	99.89	99.97	99.89	99.99	99.59	99.43	100	99.51	99.97	93.31	94.1	99.98	93.71	99.99	76.92	52.63	100	62.5	99.98
GRU	SMOTE	99.99	99.85	99.92	99.96	99.89	99.96	99.85	99.92	99.96	99.89	99.99	99.59	99.43	100	99.51	99.97	97.55	89.61	99.99	93.41	99.99	60	78.95	99.99	68.18	99.98
	No-Re	100	99.99	100	99.98	100	99.96	99.61	99.9	99.98	99.9	99.99	99.35	99.43	99.99	99.39	99.98	95.24	95.51	99.99	95.37	99.99	86.67	68.42	100	76.47	99.98

Table 3. Summary result of individual class classification performance and average mean classes detection rate of both resampled and non-resampled data for UNSW-NB15 dataset

	Classes	Analysis					Dos					Exploit					Fuzzers					Generic				
Algorithm	Metrics	De (%)	P (%)	Se(%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)
LSTM	SMOTE	88.6	101	0.26	97.59	0.43	88.99	8.03	2.03	97.68	3.24	65.58	31.61	76.27	63.2	44.69	73.41	7.87	13	80.97	9.8	82.06	97.27	55.15	99.02	70.39
	No Re	89.18	29.14	18.57	95.78	22.68	79.96	8.42	16.9	85.08	11.24	64.13	29.74	87.29	59.59	44.37	81.27	24.23	30.86	87.69	27.15	83.02	96.87	54.57	99.01	69.81
KNN	SMOTE	82.21	6.24	8.1	88.94	7.05	87.01	4.7	2.62	94.99	3.36	63.53	30.48	87.81	58.5	45.26	84.32	23.32	11.46	94.63	15.35	81.87	97.81	52.83	99.29	68.61
	No-Re	87.21	27.01	12.29	96.07	16.9	74.41	3.18	6.43	81	4.26	49.27	22.6	90.02	41.54	36.14	82.36	27.79	13.49	94.04	18.17	69.35	86.34	23.07	97.76	36.41
GRU	SMOTE	84.46	16.47	8.7	94.28	11.38	74.98	5.01	8.39	82.37	6.27	52.01	21.37	57.7	50.69	31.19	76.04	18.88	18.08	86.28	18.47	62.54	51.96	33.03	80.94	40.39
	No-Re	87.57	0.8	0.19	97.34	0.31	72.12	3.15	8.19	77.77	4.45	52.65	24.35	91.92	45.09	38.51	83.83	28.58	10.41	95.77	15.26	76.4	90.19	45.48	96.74	60.47

Table 3. Continued

	Classes	Normal					Worms (class of interest)					AvgAc (%)
Algorithm	Metrics	De (%)	P (%)	Se(%)	Sp (%)	fl (%)	De (%)	P (%)	Se (%)	Sp (%)	fl (%)	
LSTM	SMOTE	69.74	98.49	53.87	98.51	69.65	75.72	2.62	32.54	76.56	4.85	77.73
	No Re	71.95	99.09	54.61	99.21	70.41	97.64	55.66	0.3	99.89	0.57	81.02
KNN	SMOTE	66.54	84.09	54.27	84.75	65.97	87.2	4.23	23.18	88.59	7.15	78.95
	No-Re	63.32	93.15	49.8	92.18	64.9	97.04	14.06	0.9	99.84	1.68	74.71
GRU	SMOTE	57.82	92.4	42.97	91.89	58.66	78.3	4.34	35.52	79.43	7.74	64.45
	No-Re	63.83	99.26	44.7	99.38	61.64	96.55	1.96	0.6	99.18	0.92	76.14

3.3.1 Confusion Matrix: Is a good and simple matrix used in this study as it gives a general overview of model performance. The model was trained on the full development training set for UNSW-NB15 dataset, 75% full development training and scores are computed on the full evaluation set for UNSW-NB15 dataset and 25% full evaluation set. The KIDDCup99 confusion matrix of the RNNs variants shows the predicted and the actual true binary classifications of normal/attack and detection of all the four attacks group for each of the RNNs as represented on Figure 3 for multi-class task and 4 for binary task classification. Visual observation of Figure 2 shows a clear picture of the number of instances of each classes within the individual dataset. In KDDCup99, the R2L, U2R and Probes are shown to have lower connection records in Figure 2 (B), while normal and DOS appear to have more connection records. Those group with more records are learnt properly without confusing their identity while those with fewer connection records during training did not show good true positive rate and precision as it was had to identify them. This indicates data imbalance distribution problem. The dataset contains many examples for "neptune" that belongs to DOS attack class, "satan" attacks that belongs to Probe and "normal" but fewer examples of the others. Likewise, within UNSW-NB15, Figure 2 (B & C) shows that Worm has the lowest connection record than the rest of the classes within the training and testing sets, hence indicating data imbalance distribution problem.

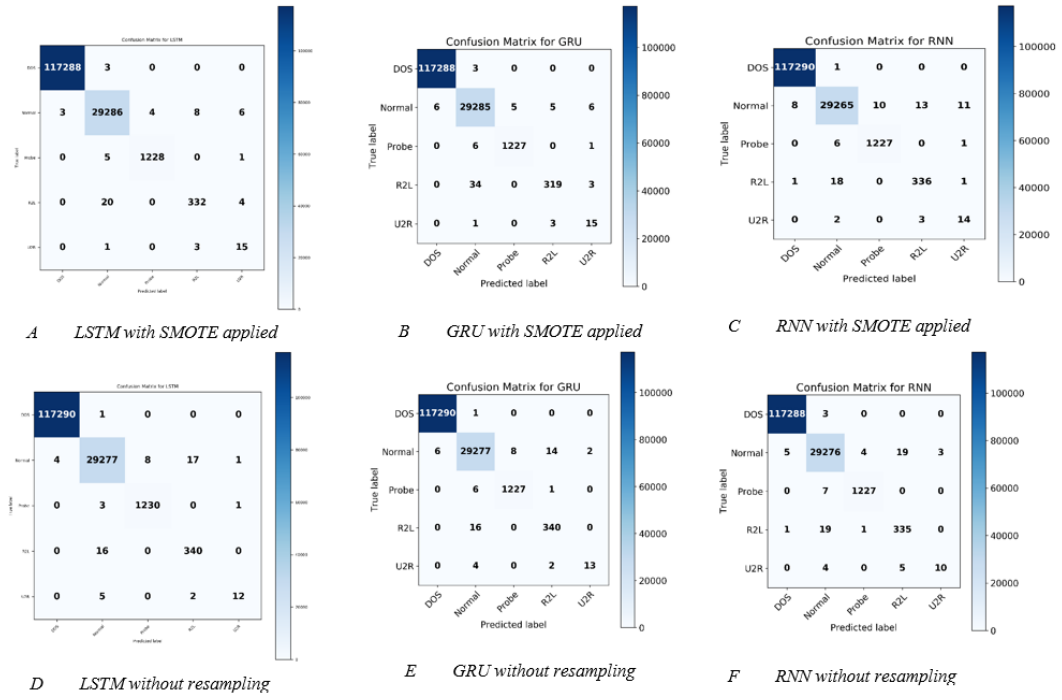


Figure 3: 10% KDDCu99 dataset confusion matrix of 5 classes with and without SMOTE oversampling techniques applied

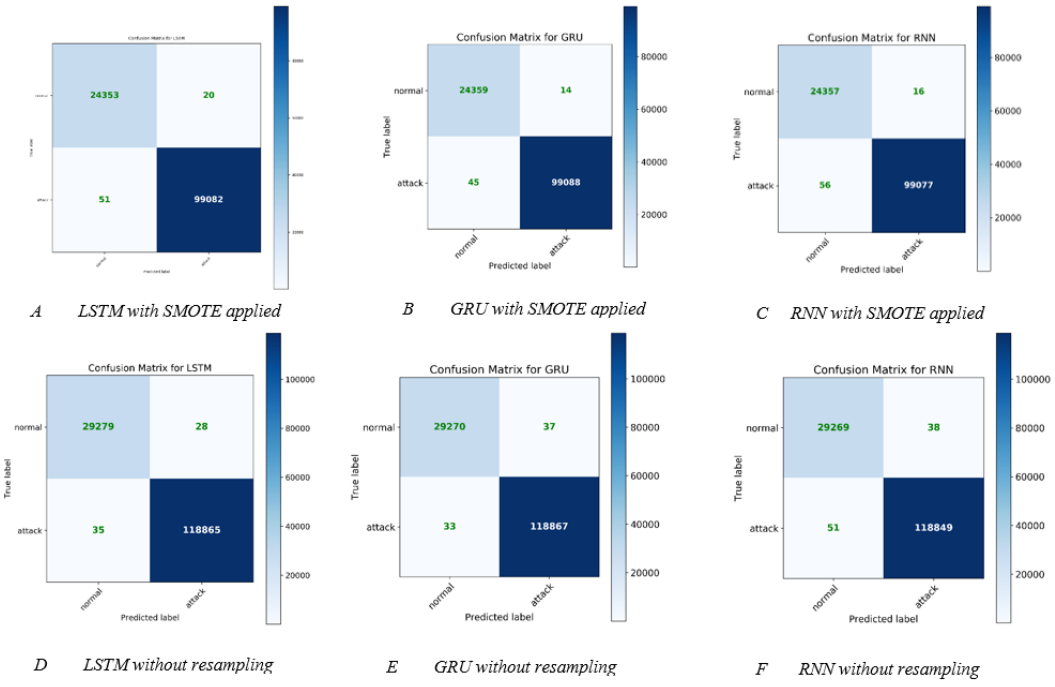
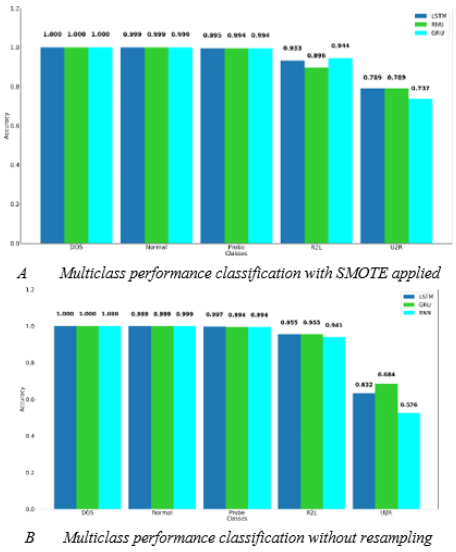


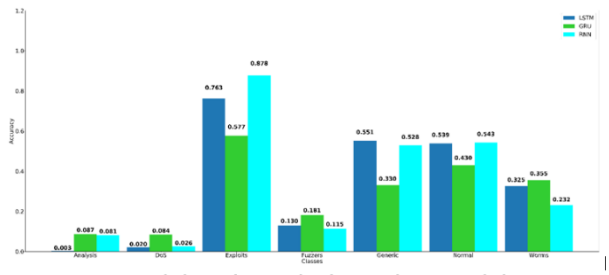
Figure 4: 10% KDDCu99 dataset confusion matrix of 2 classes with and without SMOTE oversampling techniques applied



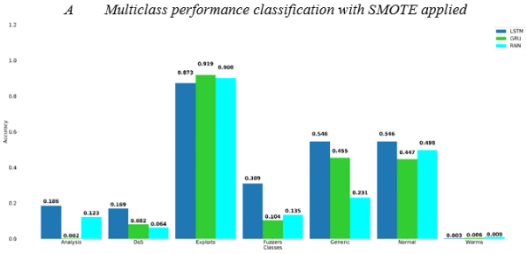
A Multiclass performance classification with SMOTE applied

B Multiclass performance classification without resampling

Figure 5a. Multi-class classification on KDDCup99



A Multiclass performance classification with SMOTE applied

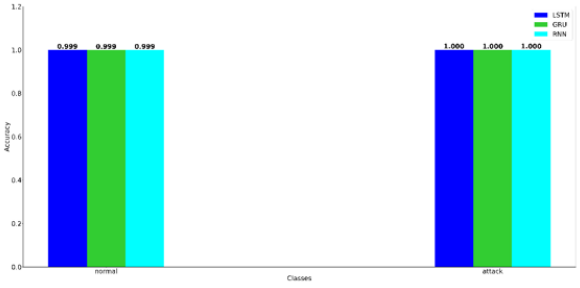


A Multiclass performance classification with SMOTE applied

Figure 5b. Multi-class classification on UNSW-NB15

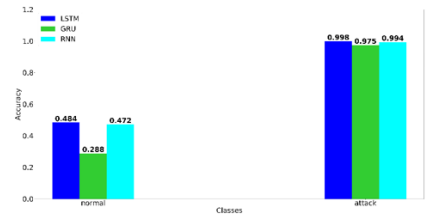
Figure 5(a & b): The visual performance representation of each of the algorithms in detecting all the four attacks family and normal network traffics on both KDDCu99 and UNSW-NB15 datasets, each variant showing prediction of each of the individual classes with and without SMOTE oversampling techniques applied. Starting from left is LSTM, followed by RNN and then GRU on the right for A and B of each dataset

A Binary performance classification with SMOTE applied

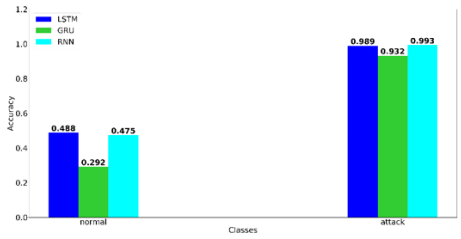


B Binary performance classification without resampling

Figure 6a. Binary classification on KDDCup99



A Binary performance classification with SMOTE applied



B Binary performance classification without resampling

Figure 6b. Binary classification on UNSW-NB15

Figure 6(a & b). The binary visual performance representation of each of the algorithms in detecting, differentiating and classify an attack and normal classes on both KDDCu99 and UNSW-NB15 datasets, each variant showing prediction of normal against attack classes with and without SMOTE oversampling techniques applied. Starting from left is LSTM, followed by GRU and then RNN on the right for both A and B of individual dataset

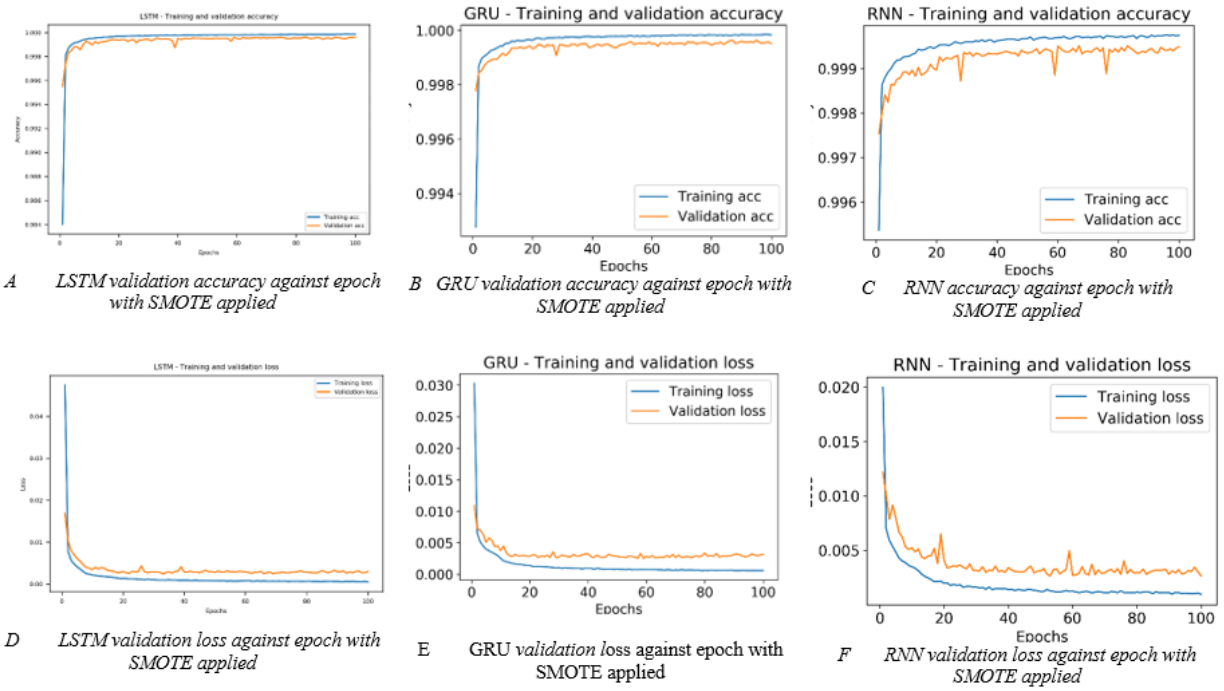


Figure 7: The visual representation of each algorithm's validation accuracy and loss rate on each iteration on 10% KDDCup99 dataset with SMOTE oversampling techniques applied.

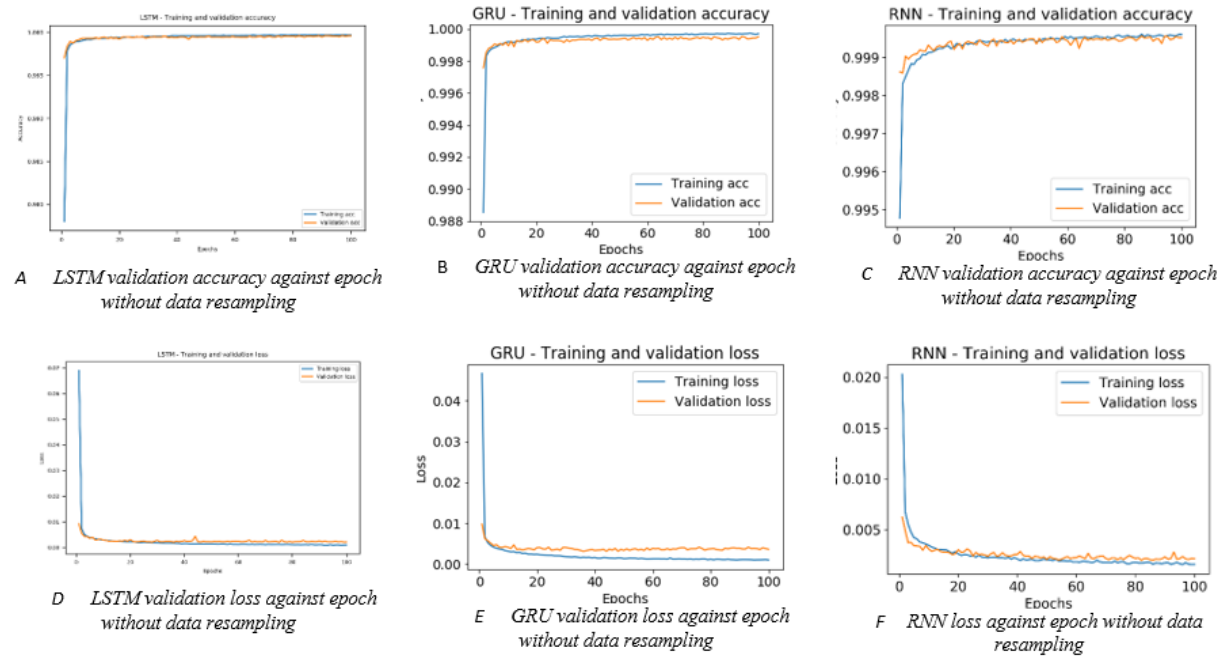
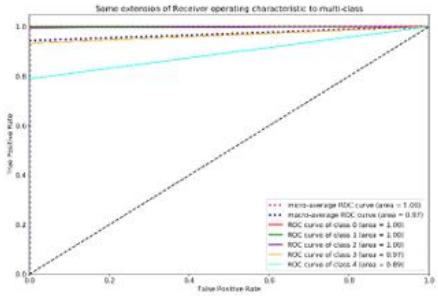
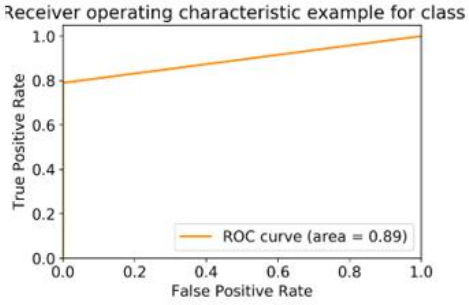


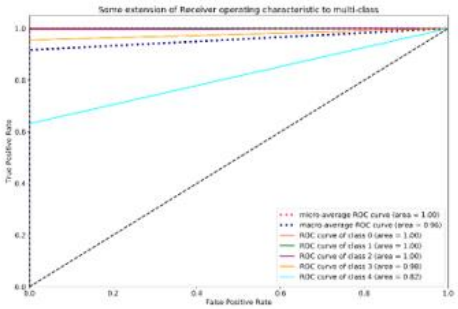
Figure 8: The visual representation of each algorithm's validation accuracy and loss rate on each iteration on 10% KDDCup99 dataset without resampling techniques applied.



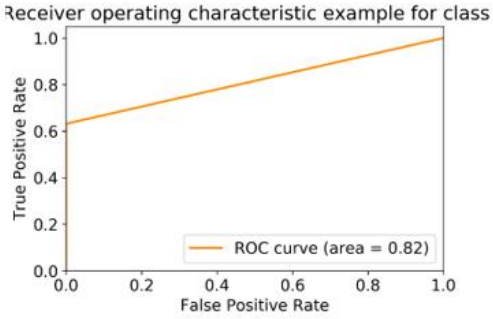
A LSTM AUC-ROC curve with SMOTE applied for all 5 classes



B LSTM AUC-ROC curve with SMOTE for class 4 (U2R)

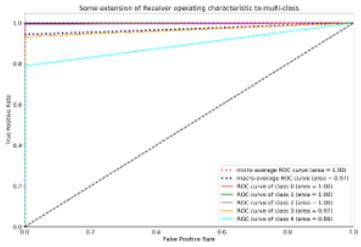


C LSTM AUC-ROC curve without resampling for all 5 classes

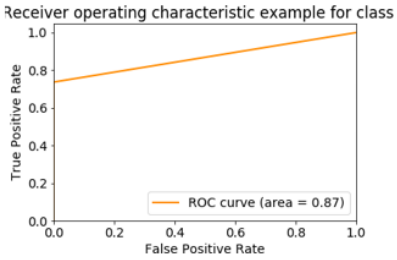


D LSTM AUC-ROC curve without resampling for class 4 (U2R)

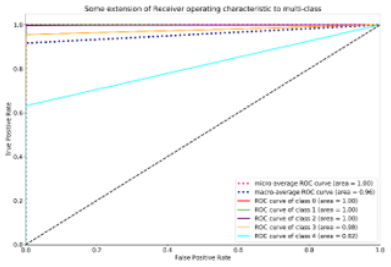
Figure 9: The visual representation of LSTM's AUC-ROC graph for all 5 classes and minority "class 4-U2R" of interest on 10% KDDCup99 dataset with SMOTE oversampling techniques applied.



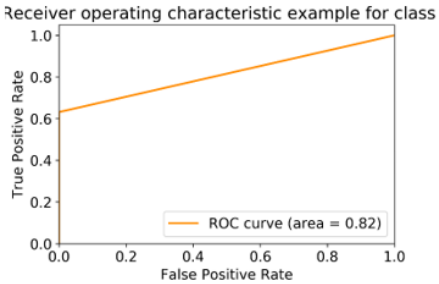
A GRU AUC-ROC curve with SMOTE applied for all 5 classes



B GRU AUC-ROC curve with SMOTE for class 4 (U2R)

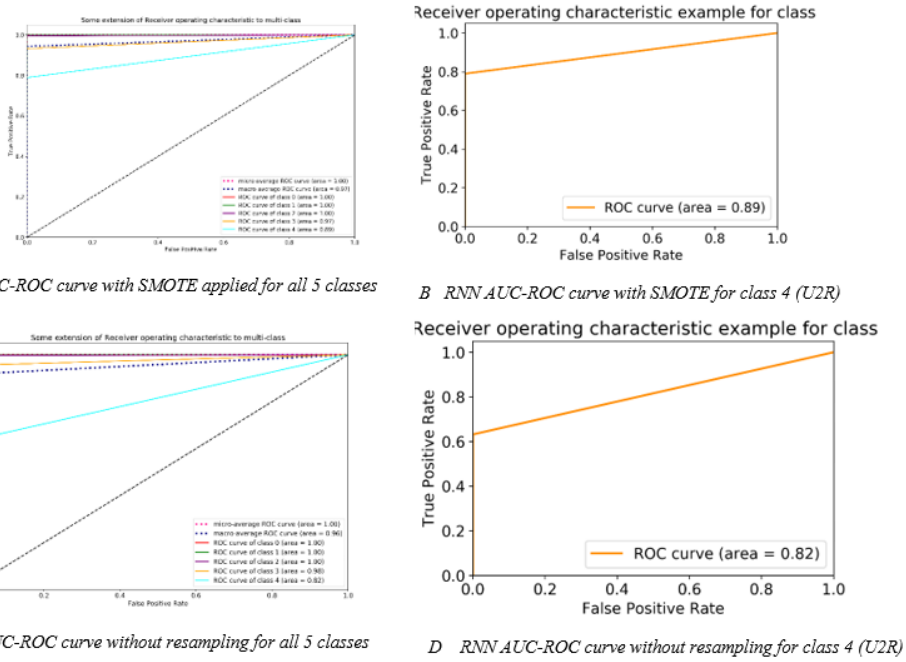


C GRU AUC-ROC curve without resampling for all 5 classes



D GRU AUC-ROC curve without resampling for class 4 (U2R)

Figure 10: The visual representation of GRU's AUC-ROC graph for all 5 classes and minority "class 4-U2R" of interest on 10% KDDCup99 dataset with SMOTE oversampling techniques applied.



A RNN AUC-ROC curve with SMOTE applied for all 5 classes

B RNN AUC-ROC curve with SMOTE for class 4 (U2R)

C RNN AUC-ROC curve without resampling for all 5 classes

D RNN AUC-ROC curve without resampling for class 4 (U2R)

Figure 11: The visual representation of RNN's AUC-ROC graph for all 5 classes and minority "class 4-U2R" of interest on 10% KDDCup99 dataset with SMOTE oversampling techniques applied

4. CONCLUSION

In this paper, SMOTE oversampling data resampling methods were implemented on two different datasets. An IDS based on heterogeneous ensemble learning approach were implemented and evaluated the impact of implementing resampling techniques and model performance on imbalance data for multi-class classification in detecting minority class of interest (rare attack). This study went further to evaluate the ability of the RNNs variants to accurately detect and classify an attack as abnormal, and how accurately this model will detect different type of attacks with our main focus on detecting the rare attack class. The result obtained from resampled data were compared to that acquired without data resampling. Based on these results as contained in Tables 1-3. It is also observed that most of the algorithms applied in this study, achieved a competitive overall average validation accuracy rate of 99.95% with 0% overall average validation loss for KDDCup99 dataset. At the same time achieving a maximum average mean accuracy of 81.02% with a noticeable validation loss for UNSW-NB15 dataset. Although the model demonstrated a good percentage detection overall average mean accuracy, precision, sensitivity and F1-score of 99.96%, and detection rate of minority class (U2R) improved from 62.50%, 56.20% and 37.50% to 78.90%, 78.90% & 73.70% when compared to our previous work in Eke et al., 2019 in classifying individual class.

Also, with the percentage macro-average obtained from both datasets, these indicates that the classifier performs well for each individual class. It can be seen that the implemented model on this study performed significantly better when dealing with KDDCup99 dataset than UNSW-NB15 dataset, especially in overall

performance validation measures. Overall, the result suggests that high classification error and class separability problem of minority class has a noticeable impact on model overall performance with respect to application domain as seen with the case of KDDCup99 and UNSW-NB15 datasets.

5. FUTURE WORK

This work on implementation of resampling techniques on diverse datasets is an ongoing study. Further work will explore the impact of implementing Principal Component Analysis (PCA) and *t*-Distributed Stochastic Neighbor Embedding (t-SNE) high-dimensional data visualisation techniques to evaluation the need in understanding the relationship boundary between classes in high-dimensional imbalance multi-class classification task and then implement OneVsOne and OneVsAll decomposition schemes together with exploring modelling combination of an optimised stacked LSTM-RNN model and CNN on a time-series dataset - UNSW-NB15 datasets (University of New South Wales 2015 Datasets) over a multi-stage APT detection architecture. As APT is a multi-step attacks, detecting a single stage of an APT technique itself does not imply detecting an APT attack as mentioned by (Ghafir, Hammoudeh, Prenosil, Han, Hegarty, Rabie, & Aparicio-Navarro, 2018). Patterns embedded in large generated datasets through industrial processes may be dynamic, hence the need for a system that can accurately detect APT in a systematic way at different time step and has the ability to learn, store and update existing patterns with the collection of new data, and also be scalable to process data in large volumes (Chen et.al, 2016). Hence, the combination of these two models to determine the efficiency of this approach since RNN variants has the capability to learn temporal dynamic behaviour over a time sequence data. The authors are currently engaged in work in this domain.

REFERENCES

Ahmad, A., Webb, J., Desouza, K. C., & Boorman, J. (2019). Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack, *Computers & Security*, vol. 86, pp. 402-418.

Akbani, R., Kwek, S., & Japkowicz, N. (2004, September). Applying support vector machines to imbalanced datasets. In *European conference on machine learning* (vol. 3201, pp. 39-50). Springer, Berlin, Heidelberg.

Ayad, A., Khalaf, M., & El-Saadany, E. (2018). Detection of false data injection attacks in automatic generation control systems considering system nonlinearities. In *2018 IEEE Electrical Power and Energy Conference (EPEC)* (pp. 1-6). IEEE.

Attenberg, J., & Ertekin, S. (2013). Class imbalance and active learning. *H. He, & Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications*, vol. 1, pp.101-149.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, vol.*6*(1), pp. 20-29.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, vol. *5*(2), pp. 57-166.

Buxton, B. F., Langdon, W. B., & Barrett, S. J. (2001). Data fusion by intelligent classifier combination. *Measurement and Control*, vol. *34*(8), pp. 229-234.

Cardie, C., & Howe, N. (1997, July). Improving minority class prediction using case-specific feature weights. In *ICML*, pp. 57-65.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority oversampling technique. *Journal of artificial intelligence research*, vol. *16*, pp. 321-357.

Chen, M. H., Chang, P. C., & Wu, J. L. (2016). A population-based incremental learning approach with artificial immune system for network intrusion detection. *Engineering Applications of Artificial Intelligence*, vol. *51*, pp. 171-181.

Chen, T, & Abu-Nimeh S. (2011). Lessons from stuxnet. Computer vol. 44(4), pp. 91– 93.

Chen, J., Su, C., Yeh, K. H., & Yung, M. (2018). Special issue on advanced persistent threat.

Chernick, M. R. (2012). Resampling methods. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2(3), pp. 255-262.

Cressey, R. (2012). Cyber Security Issues and Challenges. Booz Allen Hamilton Inc. Accessed on 2019-02-20.

Das, S., Datta, S., & Chaudhuri, B. B. (2018). Handling data irregularities in classification: Foundations, trends, and future challenges. *Pattern Recognition*, vol. *81*, pp. 674-693.

Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, vol. *41*(10), pp. 4915-4928.

Dendamrongvit, S., & Kubat, M. (2009, April). Undersampling approach for imbalanced training sets and induction from multi-label text-categorization domains. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 40-52). Springer, Berlin, Heidelberg.

Domingos, P. (1999, August). Metacost: A general method for making classifiers cost-sensitive. In *KDD*, vol. 99, pp. 155-164.

Eke, H. N., Petrovski, A., & Ahriz, H. (2019, September). The use of machine learning algorithms for detecting advanced persistent threats. In *Proceedings of the 12th International Conference on Security of Information and Networks*, pp. 1-8.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, vol. *27*(8), pp. 861-874.

Five, C. (2011). Advanced Persistent Threats: A Decade in Review. *Command Five PTY LTD*, pp. 1-13.

Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., & Aparicio-Navarro, F. J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, vol. *89*, pp. 349-359.

Giraldo-Forero, A. F., Jaramillo-Garzón, J. A., Ruiz-Muñoz, J. F., & Castellanos-Domínguez, C. G. (2013, November). Managing imbalanced data sets in multi-label problems: a case study with the SMOTE algorithm. In *Iberoamerican Congress on Pattern Recognition* (Vol.8258, pp. 334-342). Springer, Berlin, Heidelberg

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, vol. *73*, pp. 220-239.

Hoens, T. R., Qian, Q., Chawla, N. V., & Zhou, Z. H. (2012, May). Building decision trees for the multi-class imbalance problem. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 122-134). Springer, Berlin, Heidelberg.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, vol. *9*(8), pp. 1735-1780.

ISACA (2014). Advanced persistent threat awareness study results: Information systems auditing manual. Information Systems Audit and Control Association. Accessed on 2019-02-20

KP, S. (2019). RNNSecureNet: Recurrent neural networks for Cyber security use-cases. *arXiv preprint arXiv:1901.04281*.

Krawczyk, B. (2016a). Cost-sensitive one-vs-one ensemble for multi-class imbalanced data. In *2016 International Joint Conference on Neural Networks (IJCNN) (*pp. 2447-2452). IEEE.

Krawczyk, B. (2016b). Combining one-vs-one decomposition and ensemble learning for multi-class imbalanced data. In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015* (pp. 27-36). Springer, Cham.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature pp. 521.

López, V., Fernández, A., Moreno-Torres, J. G., & Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Systems with Applications*, vol. *39*(7), pp. 6585-6608.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, vol. *250*, pp. 113-141.

López, V., Del Río, S., Benítez, J. M., & Herrera, F. (2015). Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems*, pp. *258*, 5-38.

Nguyen, T. T., Nguyen, M. P., Pham, X. C., Liew, A. W. C., & Pedrycz, W. (2018). Combining heterogeneous classifiers via granular prototypes. *Applied Soft Computing*, vol. *73*, pp. 795-815.

Nguyen, T. T., Dang, M. T., Liew, A. W., & Bezdek, J. C. (2019). A weighted multiple classifier framework based on random projection. *Information Sciences*, vol. *490*, pp. 36-58.

Ma, W., Tran, D., & Sharma, D. (2008, August). Negative selection with antigen feedback in intrusion detection. In *International Conference on Artificial Immune Systems* (pp. 200-209). Springer, Berlin, Heidelberg.

McAfee Labs (2011). Cyberattacks GE. Night dragon. McAfee Foundstone Professional Services and McAfee Labs.

McDermott, C. D., Isaacs, J. P., & Petrovski, A. V. (2019, March). Evaluating Awareness and Perception of Botnet Activity within Consumer Internet-of-Things (IoT) Networks. In *Informatics* (Vol. 6, No. 1, p. 8). Multidisciplinary Digital Publishing Institute.

Micro, T. (2013). Countering the advanced persistent threat challenge with deep discovery. Trend Micro, Tech. Rep. 10.

Muñoz, A., Palomo, E. J., & Jerez-Calero, A. (2019). Use of an ANN to Value MTF and Melatonin Effect on ADHD Affected Children. IEEE *Access*, vol. 7, pp. 127254-127264.

Peng, C., Cheng, J., & Cheng, Q. (2017). A supervised learning model for high-dimensional and large-scale data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. *8*(2), pp. 30.

Ostaszewski, M., Seredynski, F., & Bouvry, P. (2006, July). Immune anomaly detection enhanced with evolutionary paradigms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 119-126). ACM.

Parikh, D., Kim, M. T., Oagaro, J., Mandayam, S., & Polikar, R. (2004, October). Combining classifiers for multisensor data fusion. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)* (Vol. 2, pp. 1232-1237). IEEE.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, vol. *6*(3), pp. 21-45.

Polikar, R., Parikh, D., & Mandayam, S. (2006, February). Multiple classifier systems for multisensor data fusion. In *Proceedings of the 2006 IEEE Sensors Applications Symposium, 2006.* (pp. 180-184). IEEE.

Sinharay, S. (2010). Resampling Method: An overview of statistics in education. *International Encyclopedia of Education*, Elsevier.

Smiraus, M., & Jasek, R. (2011). Risks of advanced persistent threats and defense against them. *Annals of DAAAM & Proceedings*, pp.1589-1591.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. *23*(04), pp. 687-719.

Sáez, J. A., Krawczyk, B., & Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, vol. *57*, pp. 164-178.

Stibor, T., Timmis, J., & Eckert, C. (2005, August). A comparative study of real-valued negative selection to statistical anomaly detection techniques. In *International Conference on Artificial Immune Systems* (pp. 262-275). Springer, Berlin, Heidelberg.

Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Evaluation of recurrent neural network and its variants for intrusion detection system (IDS). *International Journal of Information System Modeling and Design (IJISMD)*, vol. *8*(3), pp. 43-63.

Vuttipittayamongkol, P., Elyan, E., Petrovski, A., & Jayne, C. (2018, November). Overlap-Based Undersampling for Improving Imbalanced Data Classification. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 689-697). Springer, Cham.

Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM Sigkdd Explorations Newsletter*, vol. *6*(1), pp. 7-19.

Yijing, L., Haixiang, G., Xiao, L., Yanan, L., & Jinling, L. (2016). Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. *Knowledge-Based Systems*, vol. *94*, pp.88-104.

Zhang, C. X., & Duin, R. P. (2011). An experimental study of one-and two-level classifier fusion for different sample sizes. *Pattern Recognition Letters*, vol. *32*(14), pp. 1756-1767.

Zhou, Z. H., & Liu, X. Y. (2005). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, vol. 18(1), pp. 63-77.

ENDNOTES

[1] KDDCup99 dataset: Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[2] UNSW-NB15 dataset: Retrieved from https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/