# Handling Imbalanced Data in Convolutional Neural Network on Apache Spark

1st Thet Hsu Aung
*Faculty of Computer Science*
*University of Information Technology*
Yangon, Republic of the Union of Myanmar
*thethsuaung@uit.edu.mm*

2nd Aye Myat Myat Paing
*Faculty of Computer Science*
*University of Information Technology*
Yangon, Republic of the Union of Myanmar
*ayemyatmyatpaing@uit.edu.mm*

***Abstract*— With the growing pervasiveness of Big Data, many approaches have been made in several fields. The concept of big data has gained support as a result of this rising data advancement. Big data is required to show how things are related, predict future trends, and provide decision-makers with additional knowledge. Apache Spark is an effective, scalable, real-time data analytics engine that is rapidly taking over as the de facto center for big data and data science. Nowadays, how to completely collect and evaluate a huge amount of different and complicated data is a major problem. Due to these complicated data, one of the major problems is imbalanced data, which are distributed unevenly throughout the target class in datasets with; for example, one class label may contain many observations while the other may have a small number. As a result, the proposed algorithm, which is based on an oversampling method that evenly distributes the data, is used to address the problem of data imbalance. This study also emphasizes the Convolutional Neural Network (CNN), a deep learning model used with Apache Spark. The proposed system implementation is performed for balanced and imbalanced nature of NSL-KDD dataset, the benchmark dataset of intrusion detection. The experiment shows that the proposed sampling algorithm outperforms CNN models on Apache Spark compared with CNN on traditional platform.**

***Keywords—Deep Learning, Apache Spark, Oversampling, Class Imbalance***

## I. INTRODUCTION

In recent advancements, the trends of data mining and many other machine learning applications are influenced by class imbalance problems. The processing of imbalanced data is processed relying upon two categories, such as processing the data directly and through algorithms. In data mining, the imbalance of data plays a major role and is considered one of the topmost problems to be resolved. In the case of binary problems, there is only one majority and minority class. The class imbalance problems are fraud detection, software defect prediction, cancer detection. The researchers developed a reliable technique for handling the problems that occur due to class imbalance, which affects classification performance.

Due to its capacity to extract relevant features from vast amounts of data, the machine learning branch known as "Deep Learning" (DL) has recently attracted a lot of interest from the intrusion detection system (IDS). Numerous studies have demonstrated that it significantly outperforms conventional techniques and boosts the effectiveness of attack detection. The deep neural network (DNN), convolutional neural network (CNN), long short-term memory (LSTM), recurrent neural network (RNN), and other deep learning-based techniques are employed in IDS.

Even though deep learning techniques are improving IDS, they have difficulty identifying attacks with lower traffic because of the class imbalance issue. Many of the most recent benchmark IDS datasets are imbalanced when compared to real network traffic. The deep learning model cannot learn the minority classes due to the imbalanced data. As a result, the IDS's performance declines, leading to a high false alarm rate and a low detection rate, making it difficult to detect the minority threat classes. The issue of data imbalance has received too little attention in recent IDS work. All attack traffic should, however, be able to be recognized by an effective IDS. This paper uses a deep learning approach to address the issue of class imbalanced for the imbalanced class in the NSL-KDD dataset and proposes a combination of balanced algorithm with deep learning model is implemented on Apache Spark Framework.

As more data are becoming available, this is the new challenges in acquiring and processing the data to extract knowledge and analyze the effect on intrusion detection. The big data classification is handled using the spark architecture. When the volume of data is expanding quickly, it is crucial to develop tools that can tackle that data and extract value from it. Therefore, Big Data Analytics has become prominent for analyzing and managing huge amounts of data. Consecutively, this leads to faster and more efficient operations.

In this paper, Convolution Neural Network (CNN) is used as a learning model for classification in IDS on the big data analytics platform, Apache Spark. Data is initially gathered from the repositories and supplied into the preprocessing stage before being transmitted. The preprocessing stage removes the redundant and distorted data from the data, and the proposed sampling algorithm is used to address the imbalance issue. The balanced data is supplied into the mapper and reducer-based Spark architecture, where CNN is used to conduct binary-class classification. The main objective is to develop high performance and scalable IDS on big data analytics platforms, to minimize computational and storage costs, to provide class balance by applying proposed sampling methods and to analyze the system with deep learning algorithm. This paper proposes to extract valuable data from large-scale data in an effective manner and applied for class balance in the age of Big Data Analytics, scalable and high-performance predictive analytics for enormous volumes of data and high velocity of intrusion data.

By using a deep learning model with data sampling methods applied to the dataset, this paper addressed the issue

of class imbalances. Implementing the proposed sampling algorithm, which generates new samples from the minority classes, results in oversampling. On the benchmark NSL-KDD dataset, the proposed sampling algorithm is implemented as a deep learning model. The NSL-KDD intrusion detection dataset is the most used, although it also has a class imbalances issue. The model performance is evaluated by significantly obtaining performance metrics of the proposed model.

This paper is constructed as follows: Section 2 presents the related work. In section 3 proposes background theory, Section 4 presents the proposed system and section 5 shows the experimental results.

## II. Related Work

Today, daily operations are now improved by the widespread use of interconnection and interoperability of computing systems. In [13], using a deep neural network technique, this study proposed an innovative intrusion detection system with excellent network performance to identify unknown attack packages. Additionally, in this model, attack detection is carried out in two ways. A. Abdelkhalek et al. [1] established a data resampling technique for mitigating the class imbalance problem by using the Adaptive Synthetic and Tomek Links algorithms in combination with different deep learning models.

S. Jiaming et al. [2] proposed a network intrusion detection model based on two-layer CNN and Cluster-SMOTE + K-means algorithm (CSK-CNN) to process imbalanced dataset and they have been evaluated to show their accuracy, training time and testing time. In [3] proposed a hybrid strategy to address the imbalance issue to increase the minority class's detection rate while maintaining effectiveness. To reduce noise, this method combines oversampling with Synthetic Minority Over-Sampling (SMOTE) and Tomek link, an undersampling technique. Their experimental results using NSLKDD dataset showed that the overall accuracy.

Training with imbalanced data is a difficult task problem so J. M. Justin et al. [10] is used to evaluate data sampling strategies for treating high class imbalance with deep neural network and big data to solve these problems. B. G. Vikas et al. [7] engaged the data imbalance problem by conquering Synthetic Minority Oversampling Technique (SMOTE) technique that equally distributes the data. The classification is executed by employing the Apache Spark with Convolutional Neural Network.

## III. Background Theory

In this section, Apache Spark, Convolutional Neural Network, and imbalanced data of NSL-KDD are described. Big data, cloud computing, internet of things (IoT), and modern technology are all becoming more common than ever. Networks are growing in size as a result, making it harder to assess the risk of intrusions. Due to these attacks, the confidentiality, integrity, and availability (CIA) of security controls on computer or network resources are violated, resulting in an intrusion, which is defined as unauthorized access to the system. In order to protect networks from assaults, intrusion detection systems (IDS) must have a system that can identify them.

Therefore, modern systems increasingly generate massive amounts of data, driving the desire and necessity to have algorithms that can learn from this data. Dataset may easily be larger than is possible to store on a single machine, or data may arrive continuously as an infinite stream, requiring rapid processing. For this much data, distributed and parallel processing is essential which has led to an increased focus in big data platforms.

### A. Apache Spark

Apache spark is a distributed computing framework and is applied on large scale high dimensional dataset. The programming model of Apache Spark is typically based on processing resilient distributed datasets (RDD). The ability of RDD is split the computing tasks, it typically managed in a single thread way over multiple cluster node. Data should be placed in Hadoop Distributed File System (HDFS) to accomplish scalability benefit. Distributed implementation is used for many machine learning techniques, including dimensionality reduction, collaborative filtering, regression, and classification. Additionally, they enable tools for building, fine-tuning, and assessing Machine Learning pipelines in addition to loading and retaining algorithms, pipelines, and models.

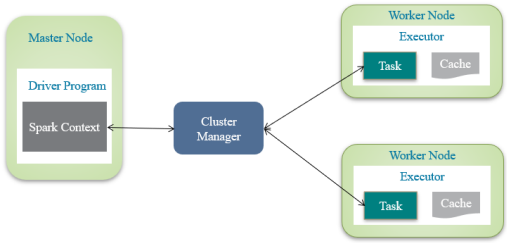


Fig. 1. Apache Spark Architecture

The Spark Cluster's architecture is depicted in Figure 1. A Spark Context object is created in the driver node and will interact with the cluster manager. The Spark Context object asks the cluster manager for resources when a Spark job starts, and the cluster manager subsequently gets executors on the worker nodes. The executors then complete their jobs by distributing part of their results to other executors on the same or different worker nodes. The driver node transmits information to the worker nodes, who subsequently relay the prepared results back to the driver node.

RDD can run in parallel with an already-existing driver application. Each time an action may be performed on an RDD, it may be recomputed. Additionally, RDDs can be duplicated across many nodes or persist on disk. As a result, adding a deep learning model to the Spark architecture increases the model's effectiveness.

### B. Convolutional Neural Network

By applying filters to groups of data points, CNN is mostly used to extract features from topology. This is widely utilized in image processing, speech recognition, time series analysis, and other fields since it may be used to record sequential as well as spatial data. It starts with an input layer

and then moves on to convolutional, pooling, and output layers. The convolution operation is performed in the convolutional layer, where filters are applied to the data points in a neighborhood and the results are carried over to the following year. CNN architecture is shown in Figure 2. Simply described, the filter is a matrix that is multiplied by the input matrix and primarily has two features, namely weights and shape. The shape represents the coverage of that filter, and the weight is something the model learns as it is being trained.

Data subsampling is done using the pooling layer. The dimensions are decreased through pooling, which lowers calculation costs. After the convolutional layer outputs the data, it collects the information and outputs the data in accordance with the pooling method selected. Additionally, it addresses the convolutional layer in CNN that is particularly important in the overfitting issue. As all the values inside the pooling window are reduced to a single value, the model is less prone to overfitting and only outputs some data while ignoring the rest.

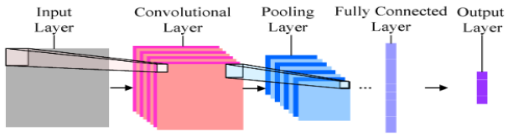


Fig. 2. Convolutional Neural Network Architecture

The CNN features are supplied to dense or completely linked layers in fully connected layers. The equation (1) describes the relationship between a perceptron's input and output in the Multi-layer Perceptron (MLP).

$$O_j = \mathrm{f}(\sum_{k=1}^{n} i_{j-1}^{k} W_{j-1}^{k}) \qquad (1)$$

Where, $O_j$ stands for the output of the $j^{th}$ perceptron in a neural network where function f is an activation function. The activation function receives input from the previous layer which is a summation of input of $(j-1)^{th}$ layer i multiplied with respective weight W.

### C. *Imbalanced Data*

When some classes are significantly underrepresented when compared to others, a dataset is said to be imbalanced. This unequal distribution reduces the effectiveness of learning algorithms by lowering the detection rate, particularly in predicting minority class like network traffic in the real world. The difficulty of detecting the minority class reduces the intrusion detection system's performance. Like in a real-world network, the NSL-KDD dataset features a considerably greater proportion of normal samples than attack samples. It is an evaluation benchmark dataset for intrusion detection systems. It is possible to think of the task of correctly classifying intrusions from network traffic as a classification problem. With the aim of accurately recognizing intrusions while reducing the false alarm rate and enhancing detection rate, a classification model can be constructed. As a result, one of the most prevalent problems in intrusion detection, class imbalances appear. The imbalanced problem in intrusion detection is still on the list of issues that need to be addressed and is mainly unresolved. Additionally, categorization of imbalanced data sets performs poorly, and small type datasets have much lower detection rates.

In order to handle binary classification challenges of imbalanced datasets in large-scale network intrusion detection, this paper's primary objective is to address them. Due to the uneven distribution of normal and abnormal samples, categorizing network traffic is fundamentally an imbalanced classification problem in the real-world network environment. This paper proposes the sampling algorithm based on oversampling algorithm. This algorithm deals with class imbalanced datasets of NSL-KDD implemented on Apache Spark and then uses CNN model, and finally make the training sample classes balanced. This paper uses accuracy, loss and training time and testing time to evaluate the proposed sampling method and compares the performance of sampling CNN model and CNN on Apache Spark.

### D. *Oversampling*

On classification datasets with an imbalanced class distribution, machine learning approaches frequently perform poorly or give falsely optimistic results. For each class to perform well, numerous methods are thought to be used on classification data with an equal number of observations. To balance or improve the class distribution in a training dataset, data sampling offers a variety of strategies. In order to balance the sample sizes for the minority class and the majority class, the oversampling technique involves increasing the number of minority class samples by randomly copying the minority samples. It modified the distribution of a variable in dataset by increasing the number of observations that take on a particular value or range of values for that variable.

The technique of oversampling method works by duplicating existing entries that are already present in the dataset to increase the presence of those entries. It facilitates the work of learning directly related to the minority class and keeping the training dataset constant. It then trains by lowering the misclassification cost in loss function, which is equal to oversampling. A technique known as class classification looks at positive things as opposed to differentiating between two outcomes. After the oversampling procedure, the data is rebuilt, and the processed data can be classified using a variety of strategies. The disadvantage of oversampling methods can increase the model overfitting to the training data. So, this paper considers this important fact and handles imbalanced dataset and studies its effect on classification.

## IV. Proposed System

This study uses the proposed sampling approach and Apache Spark architecture to manage the imbalanced data in the intrusion detection data. In this section, Figure 3 depicts the key steps of the proposed method. The imbalanced dataset NSL-KDD dataset was initially gathered from the repository. Firstly, in order to provide only the useful information, the data is first transformed into a format that is usable by eliminating the redundant data and is preprocessed to eliminate redundant data and solve concerns with data imbalance.

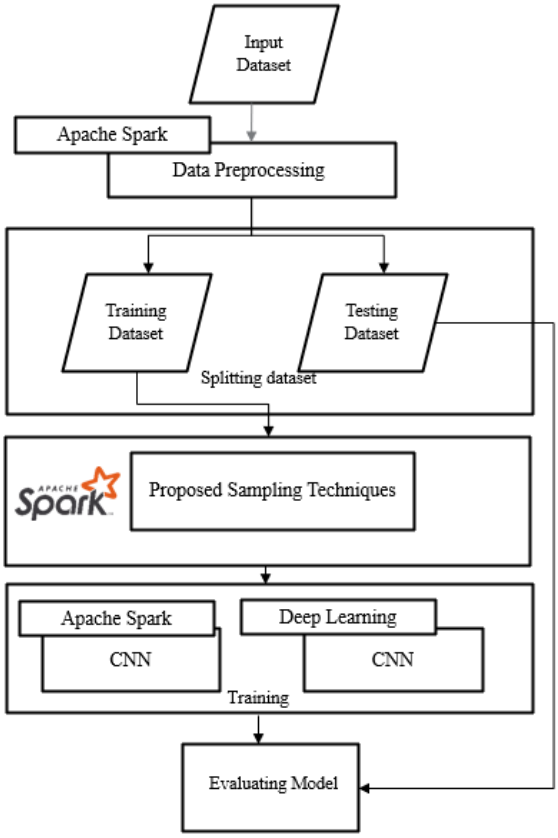


Fig. 3. Framework for the Proposed System

### A. Dataset Description

The first step is data exploration of the training dataset. The NSL-KDD dataset includes The KDDTrain+, KDDTrain+ 20%, KDDTest+, and KDDTest 21 datasets are included in the publicly available NSL-KDD dataset. Two of them for training the model, and the other two for testing the model. In the NSL-KDD dataset consists of 42 features, 41 features grouped into four categories, such as essential features, content features, time-based, and host-based features. The last feature is about a label that is maybe normal or an attack. There are five labels, one label for the normal traffic, while the attacks are grouped into four different labels (Denial of Service Attack, Probing Attack, Remote to Local Attack and User to Root Attack). Due to the imbalanced class in this dataset, it is challenging to predict a category using origin class label. According to their features, the records in this experiment's dataset are sorted into two classes: normal and abnormal. The normal class makes up the majority of classes, while the abnormal class makes up the minority. So, in the training and testing phases, the distribution of each class is divided into train and test instances.

### B. Data Preprocessing

To make the data more suitable for the classifier and address the issues with data imbalance, data preprocessing is carried out. In this step, feature preparing, and feature scaling are carried out. A critical step in the pre-processing stage is feature scaling. Data preparation typically involves the normalization procedure. The goal of normalization is to scale down the numerical values in the dataset's columns without losing any information or distorting the ranges of values. As stated in equation (2), the normalization is carried out by dividing each value by the range (Max value-Min value) and subtracting each value from the minimum value in its respective column, called Min-Max scaling.

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$

By shifting and rescaling values to fall between 0 and 1, normalization is a scaling technique. Here, the preprocessing is carried out with the proposed sampling method, which helps resolve the imbalance issues.

### C. Proposed Sampling Algoritm

The proposed sampling algorithm is presented in table 1. The first step of the proposed sampling algorithm is to load the original imbalanced dataset. And then counting the number of rows of majority class and minority class from training data (y_train), which are got as label normal defined as label1 and abnormal defined as lable2. After that the number of rows of majority class is defined as label1_amount and compute the required number of rows of minority class, which is denoted as label2_amount, for the creating of new features and create the new features of minority class starting from the last column.

Finally, until the necessary number of rows of features in the minority class is reached, build new features for the minority class. It has been demonstrated that the proposed approach enhances intrusion detection system performance. Following sampling, binary classification problems are used to train the models. The NSL-KDD test dataset is used to evaluate the models' performance and one of the most significant techniques for overcoming the challenges of big data categorization is the use of the Spark distributed computing framework to create Deep Learning models like the Convolutional Neural Network (CNN).

TABLE I. PSEUDO CODE FOR THE PROPOSED SAMPLING ALOGIRTHM

Proposed Sampling Algorithm

```
Input: Training data - x_train
Counter: count of the normal and abnormal from y_train
Output: Balanced dataset

IF label1_amount > label2_amount:
   num = label1_amount - label2_amount
   FOR i=0 to num:
        CreateFeature = Onlyabnormal[i, :]
        CreateFeature=
RemoveLastElement(CreateFeature[:len(CreateFeature)-1])
        data =  GetSingleFeature(Onlyabnormal[i + 1, -1])
        CreateFeature = AppendElement(CreateFeature, data)
        Onlyabnormal = AppendRow(Onlyabnormal,
CreateFeature)
        Y_abnormal = AppendLabelElement(Y_abnormal, 1)
   END FOR
ELSE:
   num = label2_amount - label1_amount
   FOR i=0 to num:
        CreateFeature = Onlynormal[i, :]
        CreateFeature=
RemoveLastElement(CreateFeature[:len(CreateFeature)-1])
        data = Onlynormal[i + 1, -1]
        CreateFeature = AppendElement(CreateFeature, data)
        Onlynormal = AppendRow(Onlynormal, a_del)
        Y_normal = AppendLabelElement(Y_normal, 0)
   END FOR
END IF
```

### D. Model Classifier

After that, deep learning model as CNN is trained and tested and then made prediction to classify normal and abnormal on NSL-KDD dataset and implemented the proposed sampling algorithm using Apache Spark. The two functions of map and reduce are the foundation upon which the spark architecture operates. During the map phase, the input data is split among numerous worker processes, and each worker goes through parallel processing. When the task is complete, the intermediate results are sent to the reduced phase. In this study, data processing in the mapper stage is done using CNN. The intermediate results are supplied to the reducer for defining the final class output after being shuffled using a shuffler if there is a need to group data. When possible, Spark decreases the phase or stores the results of each map in main memory. This spark architecture eliminates duplicate activities, and the process complexity is considerably decreased as a result of the removal of redundant data. Partitioned data are kept in one or more clusters.

The frequency with which classes are imbalanced is known as class imbalance. Real-world data classification problems abound, and an imbalance of classes is evident. The largest number of samples in the class-imbalanced NSL-KDD dataset. It is difficult to classify because of this severe mismatch. When datasets are imbalanced, there is a bias in categorization in favor of the majority class, which frequently results in misclassification of the minority class. Oversampling has been used to address the class imbalance. The proposed sampling algorithm is based on oversampling algorithms. The aim is to address the imbalances in the NSL-KDD dataset to increase the detection rate of minority classes while maintaining high accuracy levels. This can be achieved by addressing the issue of imbalanced classes. In this paper, rows of features take consideration of the core concept of the proposed sampling method.

## V. Experimental Results

The proposed system's performance is evaluated in this section. The performance metrics used for evaluation as presented in Figures 4, 5 and 6. The experimental results show the proposed system significantly improves the detection of minority classes and outperforms CNN on Apache Spark than CNN on traditional platform. The experimental setup is shown in Table 2.

TABLE II. SYSTEM CONFIGURATION

| | |
|---|---|
| **Operating System** | Ubuntu 20.04 LTS |
| **Host Specification** | Intel Core i7<br>8 GB RAM<br>1TB HDD |
| **VM Specification** | Virtual Box<br>6GB RAM<br>200 GB HDD |
| **Software** | Hadoop 3.2.4<br>Spark 3.2.4<br>Python 3.7 |

Below is an interpretation of the research findings that preceded the conversion of imbalanced data to balanced data using the proposed algorithm on CNN Spark architecture. The NSL-KDD is a publicly accessible dataset that can be used as a benchmark dataset by researchers to compare various intrusion detection techniques. It contains 640,672 instances. KDDTrain and KDDTest, which are divided into varying degrees of complexity, are included in the NSL-KDD dataset. The classification label appears in one column and there are 41 attributes in the dataset and the dataset's class labels are divided into two primary categories, normal and some specialized attack kinds are abnormal. There is a very clear class imbalance issue with the NSL-KDD. From this data, CNN can identify the proper features and learn the proper features. This paper employed both balanced and imbalanced data in its two tracks. Loss "categorical_crossentropy", batch sizes "32", "64", and "128", activation function "Softmax", and optimizer "Adam" are the hyperparameters for CNN in this experiment. Max-pooling approach is set up in the pooling layer with a size of pooling 2 and a stride size 2. For the training and testing percentages, the performance of the models is evaluated for the changing epochs 10, 20, 30, 40, and 50 and model accuracy is calculated according to these epochs. Due to the size of the dataset, the number of epoch values needed to correctly train the models is 50 for this experiment phase.

The impact of proposed sampling on the models can be observed and when its performance is contrasted with that of the same model when oversampling is used, its accuracy is 87.3%. The accuracy of the CNN model on Apache Spark is 97% as shown in Figure 4. According to equation 3, accuracy is the percentage of correctly classified data samples in the total amount of samples. The terms used in the confusion matrix include True positive (TP), False negative (FN), True negative (TN), False positive (FP). False positive (FP) wrongly predicts to be positive on data examples, True positive (TP) correctly predicts to be positive, False negative (FN), mistakenly predicts to be negative, and False negative (FN) wrongly predicts to be negative.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

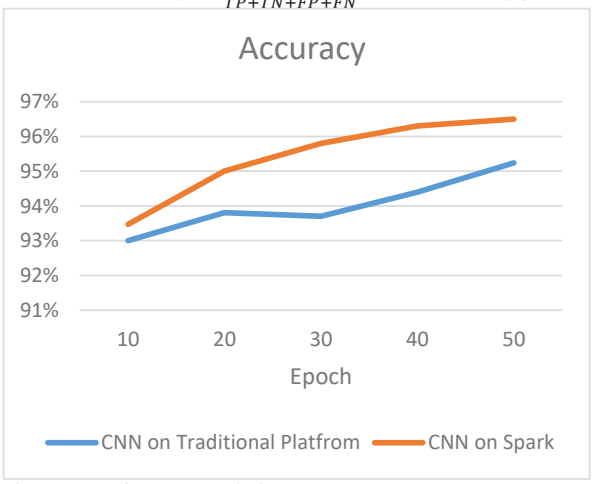


Fig. 4. Performance Analysis on Accuracy

In Figure 5 and 6 demonstrates the training and testing time for model. The CNN model using Apache Spark takes the training time by 2645.123 seconds and 127.77 seconds for testing and CNN on traditional platform takes the training time by 3060.271 seconds and 125.467 seconds for testing at epochs 50 for each. According to these results, CNN on

Apache Spark performs better than CNN for the NSL-KDD dataset.

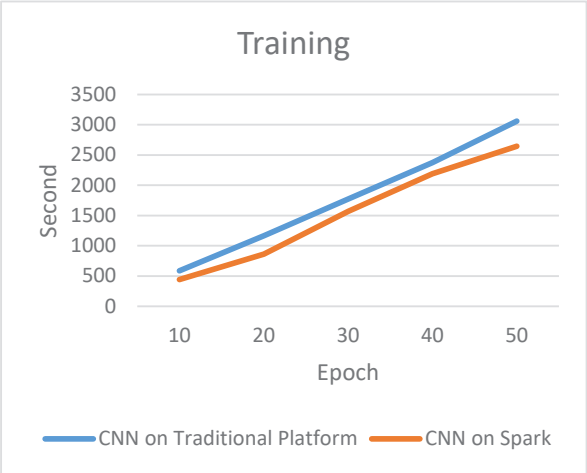


Fig. 5. Performance Analysis on Training Time

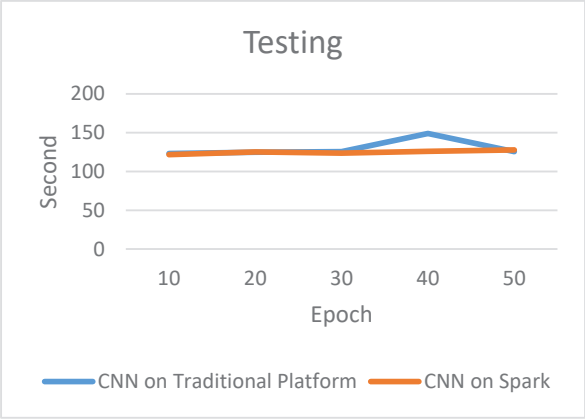


Fig. 6. Performance Analysis on Testing Time

## CONCLUSION

In this paper, the proposed system is implemented with three main modules such as data collection, data balancing and prediction. The off-line training models are created and attacked patterns are predicted. Based on this experimental result, it was discovered that the CNN model on Apache Spark obtained an accuracy of 97% when the proposed sampling algorithm was employed with deep learning models based on CNN. NSL-KDD dataset was used in this study to test CNN's deep learning model. Future research can further increase the detection rate by using various deep learning models with the proposed sampling algorithm.

## REFERENCES


[1] A. Abdelkhalek and M. Mashaly "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *The journal of Supercomputing,* pp. 10611–10644, 2023.

[2] S. Jiaming, W. Xiaojuan, H. Mingshu and J. Lei "CSK-CNN: Network Intrusion Detection Model Based on Two-Layer Convolution Neural Network for Handling Imbalanced Dataset," *MDPI. Information.*, 2023.

[3] M. Mariama and K. Hiroshi, "Handling class Imbalance problem in Intrusion Detection System based on Deep Learning," *International Journal of Networking and Computing.*, vol. 12, no. 2, pp. 467–492, Jul. 2022.

[4] X. Wenhao, L. Gongqian, D. Zhonghui, T. Baoyu and Z. Baosheng, "An Improved Oversampling Algorithm Based on the Samples' Selection Strategy for Classifying Imbalanced Data,", *Mathematical Problems in Engineering,* 2019.

[5] A. Reem, G. Mossa, O. Saud and A. Faisal, "Intrusion Detection Model for Imbalanced Dataset using SMOTE and Random Forest Algorithm,", *International Conference on Advances in Cyber Security,* pp. 361–378, Jan. 2022.

[6] B. Sikha and L. Kunqi, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data.*, 2021.

[7] B. G. Vikas and R. Hanumantha, "Spark-based deep classifier framework for imbalanced data classification," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization,* Feb. 2023.

[8] G. Anand, T. K. Hardeo, S. Ritvik, K. Pulkit and N. Sreyashi, "A Big Data Analysis Framework Using Apache Spark and Deep Learning," *International Conference on Data Mining Workshop,* Nov. 2017.

[9] J. M. Justin, and K. M. Jaghi, "The Effects of Data Sampling with Deep Learning and Highly Imbalanced Big Data," *Information System Frontiers,* Jun. 2020.

[10] C. William, and K. Bartosz, "Imbalanced Big Data Oversampling: Taxonomy, Algorithms Software, Guidelines and Future Directions," Jul. 2021.

[11] F. Osama and D. Erdogan, "Intrusion Detection Using Big Data and Deep Learning Techniques," *ACM Southeast Conference - ACMSE,* April, 2019.

[12] A. Lirim and D. Cihan, "Network Intrusion Detection System using Deep Learning," *Complex Adaptive Systems Conference Theme: Big Data, IOT and AI for a Smarter Future,* Jun, 2021.

[13] S. V. Spelmen, and R. Prokodi, "A Review on Handling Imbalanced Data," *International Conference on Current Trends toward Converging Technologies,* 2018.