Performance Measurement of Convolutional Neural Network Using Knowledge Discovery in Databases (KDD) Dataset

Tinuke Omolewa Oladele[1] and Ifetayo Remilekun Agboola[2]
[1,2]Department of Computer Science, Faculty of Communication and Information Sciences,
University of Ilorin, Ilorin, Kwara State,
Nigeria.
Email: [1]oladele.to@unilorin.edu.ng
[2]oyat2010@gmail.com

ABSTRACT
With the increased use of internet across various devices, for various activities, the demand for security across networks has only been increasing. Several activities of abuse, misuse and network policy violators are also on the increase. There is need to advance the level of security to check unethical use of systems and network resources. The mode by which attackers infiltrate the network have become dynamic. Many Intrusion Detection Systems (IDS) have been developed, but have experienced the problem of false positive or false negative alarms. In this paper, a method using convolutional neural networks to classify attacks while increasing accuracy and performance is proposed. An example over the Knowledge Discovery in Databases (KDD) Dataset is detailed in this study to illustrate the effectiveness of the proposed methodology.

Keywords: Convolutional Neural Network, Firewall, Hacker, Intrusion Detection Systems (IDS), Machine Learning

African Journal of Computing & ICT Reference Format:

Tinuke Omolewa Oladele & Ifetayo Remilekun Agboola (2017), Performance Measurement of Convolutional Neural Network using Knowledge Discovery in Databases (KDD) Dataset, *Afr. J. Comp. & ICT*, Vol.10, Nos. 1& 2, pp. 24 - 36.

I. INTRODUCTION

The world is now driven by data over the internet through social networks, e-commerce, online banking and so on, and all these information are stored or managed via the internet and automated processes carried out through the use of Information Technology (IT) systems. Advances of new tools, methods, and cyber-attack is steadily growing in terms of number of attacks and degree of damage caused to its victims. It was discovered that hackers are developing new ways to gain illegal entry to networks, programs and records. The intention of attackers is to compromise the confidentiality, integrity and availability of information, building their targets from individuals to small or medium sized businesses and even commercial enterprise [1].

Asia Pacific Computer Emergency Response Team (CERT) annual report of 2016, presented a remarkable rise in the number of intrusions and cyber-security threat in the last decade [2]. Likewise, according to [3] an

account from the Malaysia CERT published in 2016, states that 43% of 9986 hateful occurrences involve intrusions during working hours. Every day seems to bring a bigger wide variety of attacks globally, but additionally a larger range of attacks defeating the security of big establishments, thus affecting the information security, business continuity and customers' trust. The growing trend has reached new peak in 2016, universally known as "the year of cyber-attacks", but the authors believe this is not to be the peak except countermeasures are kept in place [4].

An intrusion is an unwanted or unsuspected entry or attack into a device, system or network. IDS are devices or software applications that monitor networks or devices to identify these attacks, report them and to correct them in some cases. They can be a part of a specific device communicating with the OS (known as Host based IDS) or a part of a network (NIDS). The detection can be based on known patterns of behaviour or data exchange (signature based) or unknown patterns (anomaly based). In general, these systems are rule based, and use a knowledge base with known patterns and configurations. There are primarily three kinds of IDS; Anomaly based IDS, Signature based IDS, and Host based IDS. There is need to advance the level of security to check unethical use of systems and network resources. The mode by which attackers infiltrate the network have become dynamic [3] [5].

Many Intrusion Detection Systems (IDS) have been reported in the literature. However, the problem of false positive or false negative alarms is still a major issue. This research is restricted to the simulation and analyzing performance of Convolutional Neural Network procedure on the Knowledge Discovery Dataset Cup dataset from KDD 99. The KDD 99 dataset database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. The advantage of using this dataset is that the dataset can be sorted and labelled therefore one does not need to spend time generating a new dataset. Waikato Environment Knowledge Analysis (WEKA) machine learning tool was used to evaluate the performance of the algorithm. Denial of Service, probing, user-to-root and remote-to-local types of attacks in the KDDcup '99 dataset [6].

II. RELATED WORKS

A. Genetic Algorithms

Madhavi et. al. presented a genetic algorithm method that modifies itself and gets better at finding solutions by simulating the process of natural selection, in evolution, such as "survival of the fittest". In their model, features are extracted based on their usefulness. Rules are fed in two parts: condition and result (such as, if - then). Each feature is extracted in scalar numeric form and fuzzified such that it's value lies in a range. This is done using "membership functions" that translate both what and how much in each value. For example, if height is the attribute, it's values would be {tall, medium, short) and what height corresponds to tall, medium, short is defined, known as truth of membership; 180 cm is "tall" whereas 150 is "short". The simplest way to do this is using the triangular membership function, which has a graph consisting of 3 points: upper limit, lower limit and a middle value that lies between the two [7]. Like every genetic algorithm, this begins with initialization of the population [8]. It involves conversion of the rules into numeric form (in bits). Based on the "information gain" that is, "fitness score" of each feature, 15 features are chosen. The results are computed and stored in a confusion matrix for easier analysis, next to a confusion matrix of the classic genetic system. The True Positive Rate, True Negative Rate, and then the detection rate (no of true detections/ total no of intrusions). The proposed system is shown to be better than the classical system by 2.4% accuracy. Weakness of proposed model is that it does not scale with more data. The number of false negatives increase with more data.

Statistical Data Analysis

Milan and Bulatovic describes the Statistical-based Intrusion Detection System (IDS) as the behaviour of the system that is represented from a random viewpoint, activities of various subjects are analysed and a suitable profile encapsulating the behaviour generated. An abnormality value or score is assigned to each profile [9].
Advantage: It is straightforward. Abnormal behaviour may be easily defined as anything lying a few values away from the average or deviation of a function over time.

Disadvantage: This method does not consider the relationships and the chronology of the events. Intruders may also be able to monitor the network and understand what behaviour is considered abnormal and launch smarter attacks. The determination of what is to be considered abnormal is difficult.

Artificial Intelligence (AI) and Expert Systems
Artificial Intelligence (AI) and Expert systems can be used in identifying behavioural patterns behind malicious attacks and forming a rule base. Expert systems are used for this purpose in which the knowledge base consists of 'if then' conditions which may be traversed in any order to identify anomalies [9]. Disadvantage: Only known patterns can be recognized.

Bayesian Method
This method operates by identifying that all features have diverse probabilities of occurring in attacks and in normal network traffic. The filter is trained by giving it classified traffic, which then computed probabilities for each feature. After training, the filter will calculate the probabilities for each connection and classify it as either normal traffic or an attack [9].

The negatives are that it is rule based. The effectiveness of the system hangs on how well it is executed and verified on all protocols. Rule defining procedure is also affected by various protocols used by numerous operators. Apart from these, custom protocols also make rule defining a tedious job. For detection to occur correctly, the comprehensive information about the recognized network behaviour need to be established [10].

Pattern Mining
To perform the search, a large string is usually found from a list of strings by constructing a DFA and following the Aho-Corasick method. A head-body algorithm was proposed in which the AC-DFA was divided into head and body parts such that the "head" partition continue to behave like a conventional AC-DFA but the body can perform search on full body length patterns by being converted into an NFA. In this method, each string from the list is built into a trie, which has 3 main functions: 1. Follow Edges 2. Failure: Store previous edges when current value does not have an edge. 3. Output: String combinations that have occurred at current state. The trie is then made into NFA to support linear time matching, and search is pursued. Here pattern matching can also be done for lengths that are not predefined. The only weakness of this model is in the partitioning of the string: what is the idea length? [11] [12].

III. METHODOLOGY

This paper makes use of Convolutional Neural network (CNN) method to classify attacks. Convolution is a scientific idea utilized vigorously in Digital Signal Processing when managing signals that appear as a period arrangement [13]. Generally, convolution is an instrument to join or "mix" two rudiments of time. This model encompasses an input, a hidden and output layers. The number of nodes in the input layer and the hidden layers are set to the amount of the feature in the input feature path. Similarly, the number of nodes in the output layer is fixed to the number of output. Additionally, the input layer and the hidden layers have an extra biased unit whose input is a constant set of values. The preprocessing phase comprise splitting the KDD99 dataset into various types of attacks. Feature selection was performed using information gain technique along with Ranker method. KDD99 dataset database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment [6]. Ranker method rank attributes by their individual evaluations. It is used in conjunction with attribute evaluators.

A.1. **K-fold Cross Validation**
This is a mechanism in machine learning that trains and test an algorithm by dividing the dataset into random K-partitions. The K partitioned dataset was used for testing the algorithm for correct prediction while all other preceding partitions are used to train the algorithm. The first 9 partitioned datasets train the algorithm while the other groups are used for testing. The process is repeated until all groups are covered and the performance is measured as the aggregate of all the K-folds.

A.2**. Percentage Split**
Percentage Split (Fixed or Holdout) leave out random N% of the data. 90% of the rows are used for building the model and 10% for testing the model. The algorithm is trained against the trained data and the accuracy is calculated on the whole data set.

3.1 CONVOLUTIONAL NEURAL NETWORK
Convolution Neural Network (ConvNet) is an arrangement of layers and each layer of a ConvNet converts one dimension of activations to another through a differentiable function [13]. ConvNet architectures conventionally use the following parameters and they are applied in this paper.

- Input / Output Volumes Features
- CONV (Convolutional Layer)
- ReLu (Rectified Linear Unit) Layer
- Pool Layer
- Fully Connected Layer (FC)

1. **INPUT**: this is the features-selected dataset containing attributes with class labels.

2. **Convolutional layer**: In this layer neurons that are associated to local regions in the input are computed, a dot product is calculated between their weights and a small region they are connected to in the input volume. It performs the core operations of training and therefore firing the neurons of the network. It performs the convolution operation over the inputted KDD datasets. Each receptive field contains neuron that are connected to a certain region of input in the dataset.

3. **RELU**: (Rectifier Linear Unit) frequently positioned activation function intended for the outputs of the CNN.

4. **POOL**: The pooling layer is regularly positioned after the Convolutional layer. Its main function lies in reducing or perform a down-sampling operation. The reduction in size leads to less computational overhead for the upcoming layers of the network and it helps to manage overfitting.

5. **Fully-Connected**: this layer calculates the class scores, resulting in input size where each of the numbers matches to a class label.

3.2 INFORMATION GAIN IN TERMS OF FEATURE SELECTION

According to [14], prior to pre-processing of data, the features of the data set are identified as either being significant to the intrusion detection process, or redundant. Irrelevant features are usually observed to be closely correlated with one or more other features. As a result, overlooking them from the intrusion detection process does not degrade classification accuracy. In fact, it also enhances and improve the accuracy due to the subsequent data reduction, and elimination of noise and measurement errors associated with the degraded features. Therefore, selecting a good subset of features shows to be significant in improving the overall performance of the system as shown in figure 1.

3.3 DATASET

According to Yimin, KDD'99 is one of the frequently and most generally used data set for the evaluation of a detection technique [15] [16]. 10% of KDD99 dataset was used to simulate for the purpose of this work. The dataset consists of data captured in DARPA'98 IDS evaluation program [6]. DARPA'98, which is almost 4 gigabytes of compressed raw TCP dump data of 7 weeks of network traffic. 2 million data connection records tested for about 2 weeks.

3.4 TRAINING CNN WITH ALL FEATURES

The training dataset will be used to train the CNN intrusion detection system for detection and correct classification of attacks and it is subsequently tested to see if learning algorithm is able to correctly classify the attacks. This training and testing process will be carried out using 10-fold cross validation technique and percentage split.

3.5 FEATURE SELECTION VIA INFORMATION GAIN

Features denotes and suggests a function that is distinct and a useful observation or pattern obtained from the entered data that aids in execution in order to obtain better result. The CNN learns the features from the entered datasets. Usually, they occur repeatedly from the data and gain prominence. For example, while performing Face Detection, the reality that every human face has a pair of eyes may be treated as a feature by the system, as a way to be detected and learned by using the distinct layers. In generic object classification, the threshold contours of the objects serve as the features.

To achieve the aim of this research work, feature reduction of the dataset was performed via information gain. This would help in the reduction of the number of features that the CNN classifier will use in making correct and accurate classification. The feature reduction/selection technique will best select some attributes from the original 41 features (plus the label feature).

3.6 EXPERIMENT AND DISCUSSION

A. Introduction

This section gives detailed information about experiments carried out and the testing for the purpose of this work. It describes the parameters used for the implementations, testing, and gives the results of the experiments. Comparisons were made among similar experiment's results, and with the results gotten from related literature.

B. **Performance Measurement Terms**

The performance of the classification algorithms used were measured using:

- **True Positive**: is the number of detected attacks and it is indeed an attack.
- **True Negative**: amount of normal instance correctly classified as normal.
- **False Positive**: also known as false alarm. Amount of normal behaviour grouped as an attack.

- **False Negative**: quantity of normal instances grouped as an attack.
- **Training Time (TT)**: this is the time taken to build the model.
- Kappa = (observed accuracy - expected accuracy)/ (1 - expected accuracy)
- F-measure is the harmonic mean of the precision and recall.

§ = 2 * Precision* Recall / (Recall + Precision)

Simulation Tools

Experiments were performed using WEKA 3.8 [17]. It is a Java program containing a lot of data mining algorithms used for classification, clustering, feature selection/ filter, visualizing and so on.

It was developed at the University of Wakaito, New Zealand and the name is an acronym for Wakaito Environment for Knowledge Analysis. It is an opensource software available under general public license. The experiments were carried out on a 64-bit Windows Server 2012 R2 system unit with 8 GB RAM and an Intel® Xeon ® CPU @ 2.40GHz as the experiments require a resultant processing power due to its iterative nature.

IV. RESULTS

Conclusively, the result derived from the operation was subjected to analysis and comparison. The performance was evaluated and measured respectively via precision, recall, training time, kappa statistics, accuracy and many more. Basically, the true positive, false positive, true negative, and false negative of the operations as they are the variables that will be used to evaluate the performance of the algorithm.

To assess the effectiveness of the algorithm, a 10-fold cross validation of KDD data set and percentage split test was carried out in a Waikato Environment for Knowledge Analysis (Weka) environment. 10-fold cross validation and 90 % split and 10% training was used to test and evaluate the algorithms. This procedure involves separating the data set is into 10 subsets for cross validation. One of the 10 subsets was used as the test set and the other for the training set. This delivers a good indication of how good the classifier performed on unknown signatures.

The following results were obtained at the end of the simulation as shown in Table 3.

The detailed accuracy by class for cross validation is depicted in figure 2 as shown in table 3. Table 4 shows the outcome of performance of Convolution Neural Network. Figure 3 shows the total time for building the model.

4.1 DISCUSSION OF RESULTS

The performance of convolutional neural network was examined and it was classified using Precision, Recall, Accuracy, False Positive (FP), True Positive (TP), True Negative (TN), and False Negative (FN) as shown in the Table 4. Attaining high precision and accuracy was possible by choosing carefully the sample type through information gain. The performance was measured via properly classified instance (Accuracy), and incorrectly classified instance. Signature Based Intrusion detection systems will not only be rule based but it can now detect unknown attackers through the training of the known signatures and was able to classify with higher accuracy. The results are computed and stored in a confusion matrix for easier analysis. The True Positive Rate, True Negative Rate is 0.990 and 0.002, respectively and the Detection rate is 99.04%. The proposed system is shown to be better than the existing system by 1.06% accuracy.

V. CONCLUSION

Convolutional neural networks are a class of deep neural networks that share weights in their convolutional layers. With their ability to learn distinguishing features and providing high performance with improved false positive rates, they can be used in developing classifiers for Intrusion Detection Systems. The prototype discussed in the paper showed up to 99.04% accuracy, with 0.002 false positive rates and a training time of 185.41seconds.

REFERENCES

[1] Bendovschi and Andree A. (2015). Cyber-Attacks – Trends, Patterns and Security Countermeasures. *Science Direct.* Oxford.

[2] APCERT (2016). Asia Pacific Computer Emergency Response Team (APCERT). APCERT Annual Report 2016. APCERT Secretariat. URL:http://www.apcert.org.

[3] Anwar S., Zain J. M., Zolkipli M. F., Inayat Z., Khan S., Anthony B. and Chang B. (2017). From Intrusion From Detection to an Intrusion Response System: Fundamentals, Requirements,

and Future Directions. (S. L. Aung, Ed.) *MDPI*, 7-10. Accessed April 2017, from http://creativecommons.org/licenses/by/4.0/

[4] CISCO. (2017). *Cisco 2017 Cyber Security Report*. San Jose: Cisco Security Research. Retrieved 2017, from www.cisco.com/go/acr2017errata

[5] Babirye, C. (2015). Intrusion Detection Techniques in Manets. *Researchgate*, 4-5. Retrieved 1 18, 2017, from https://www.researchgate.net/publication/310616681_intrusion_detection_techniques_in_manets?enrichId=rgreq-1162bb3f8c7773d47068730b9346f415-XXX&enrichSource=Y292ZXJQYWdlOzMxMDYxNjY4MTtBUzo0MzExNDI4NDg0NzEwNDBAMTQ3OTgwMzg3MzAzMw%3D%3D&el=1_x_3&_esc=publica

[6] KDD'99 datasets (1999). The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999.

[7] Madhavi, Koustubha B., Mohan V. & Vilas M Ghodki. (2016). An Enhanced Genetic Algorithm based Intrusion Detection System for detection of Denial –of-Service Attacks. *International Journal of Computer Science Trends and Technology (IJCST), 4*(5). Accessed April 2017, from www.ijcstjournal.org

[8] Saheed Y. K. and Babatunde A. O. (2014). Genetic Algorithm Technique in Program Path Coverage For Improving Software Testing. African Journal of Computing & ICTs. IEEE 7 (5), 151 – 158.

[9] Milan T. & Bulatovic D. (2008). *Design of an Intrusion Detection System Based on Bayesian Networks* (Vol. 8). Accessed 12 June 2016.

[10] Rajan, S. M. and Sundara M. (2014). Detecting Anomaly IDS in Network using Bayesian Network. *IOSR Journal of Computer Engineering (IOSR-JCE), 16*(1), 1-8. Retrieved Feb 2017, from www.iosrjournals.

[11] Yang Y. H. E., Prasanna V. K. and Jiang C. (2010). Head-Body Partitioned String Matching for Deep Packet Inspection with scalable and Attack-Resilient Performance. Proceedings of 24th IEEE International Parallel and Distributed Processing Symposium (IPDPS).

https://zdoc.site.

[12] Aho A. V.and Corasick M. J.(1975). Efficient String Matching: An Aid to Bibliographic Search. Communications of the ACM vol. 18, no. 6, pp. 333–340, 1975.

[13] Sekuboyina A. K., Devarakonda S. T., and Seelamantula C. S. (2017). A Convolutional Neural Network Approach for Abnormality Detection in Wireless Capsule Endoscopy. *IEEE*, 1-4.

[14] Ibrahim, Badr and Shaheen. (2012). *Information gain and feature selection* (Vol. 4). Retrieved 2016.

[15] Yimin W. (2004). High-dimensional Pattern Analysis in Multimedia Information Retrieval and Bioinformatics, Doctoral Thesis, State University of New York, January 2004.

[16] Balogun, A. O. and Jimoh, R. G. (2016). Anomaly Intrusion Detection Using an Hybrid Of Decision Tree And K-Nearest Neighbor*A Multidisciplinary Journal Publication of the Faculty of Science, Adeleke University, Ede, Nigeria*, 68-73.

[17] WEKA software, Machine Learning, http://www.cs.waikato.ac.nz/ml/weka/, The University of Waikato, Hamilton, New Zealand.

[18] Hodo E., Bellekens X., Hamilton A., Tachtatzis C. and Atkinson R. (2017). Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. *Department of Electronic & Electrical Engineering University of Strathclyde*, 2-20. Retrieved 2017 from www.researchgate.net

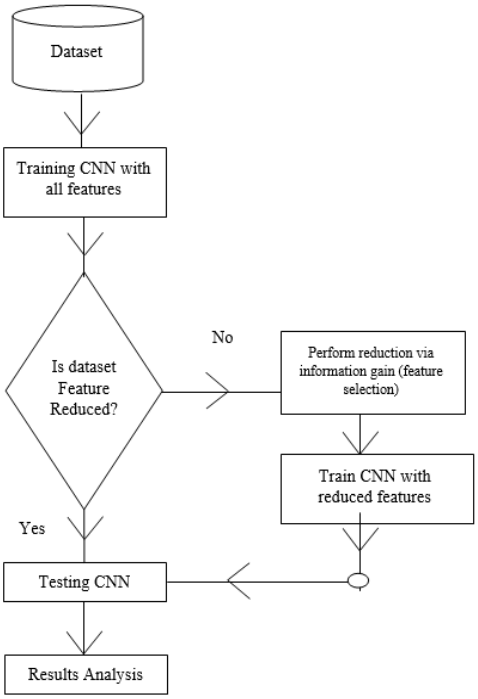

Figure 1: System Architecture

Table 1: Extracted features based on analysis of each header field

No.	Feature	Feature Description
1	IP Source	IP protocol
2	IP Destination	IP Length
3	TCP Source Port	IP Type of Service
4	TCP Destination Port	IP Header Length
5	UDP Source Port	IP Time to Live
6	UDP Destination Port	Ether Source
7	UDP Length	Ethernet Size
8	ICMP Code	Ethernet Destination
9	ICMP Type	Ethernet Protocol

Table 2: Example of features extracted from specific TCP connection

	Features	Description of Features
1	Period	Connection length in seconds
2	protocol_type	Protocol type used, example, up, tcp.
3	Service	Destination of network service, example, http, telnet.
4	scr_bytes	Amount of data bytes from source to destination
5	dst_bytes	Amount of data bytes form source to destination
6	Flag	Normal or error status of network
7	wrong_fragment	Quantity of wrong fragments
8	Urgent	Quantity of wrong fragments

Table 3: Results of Performance Evaluation of Convolutional Neural Network.

PARAMETERS	KDD'99Cup Dataset	
	Cross Validation	Percentage Split (90%)
CORRECTLY CLASSIFIED INSTANCES (%)	99.04	99.0026
INCORRECTLY CLASSIFIED INSTANCES (%)	0.96	0.9974
KAPPA STATISTICS	0.9937	0.983
MEAN ABSOLUTE ERROR	0.0016	0.0017
ROOT MEAN SQUARED ERROR	0.0268	0.0274
RELATIVE ABSOLUTE ERROR	3.1237	3.3038
ROOT RELATIVE SQUARED ERROR(RRSE)	16.7295	17.1247

Table 4: Outcome of Performance Measurement of Convolutional Neural Network.

PARAMETERS	KDD'99Cup Dataset	
	Cross Validation	Percentage Split (90%)
TP RATE	0.990	0.990
FP RATE	0.002	0.002
PRECISION	0.985	0.985
RECALL	0.990	0.990
F-MEASURE	0.988	0.987
RECEIVER OPERATING CHARACTERISTIC CURVE	0.999	1.000
PRECISION-RECALL CURVES AREA	0.992	0.993
TRAINING TIME	185.41secs	142.17secs

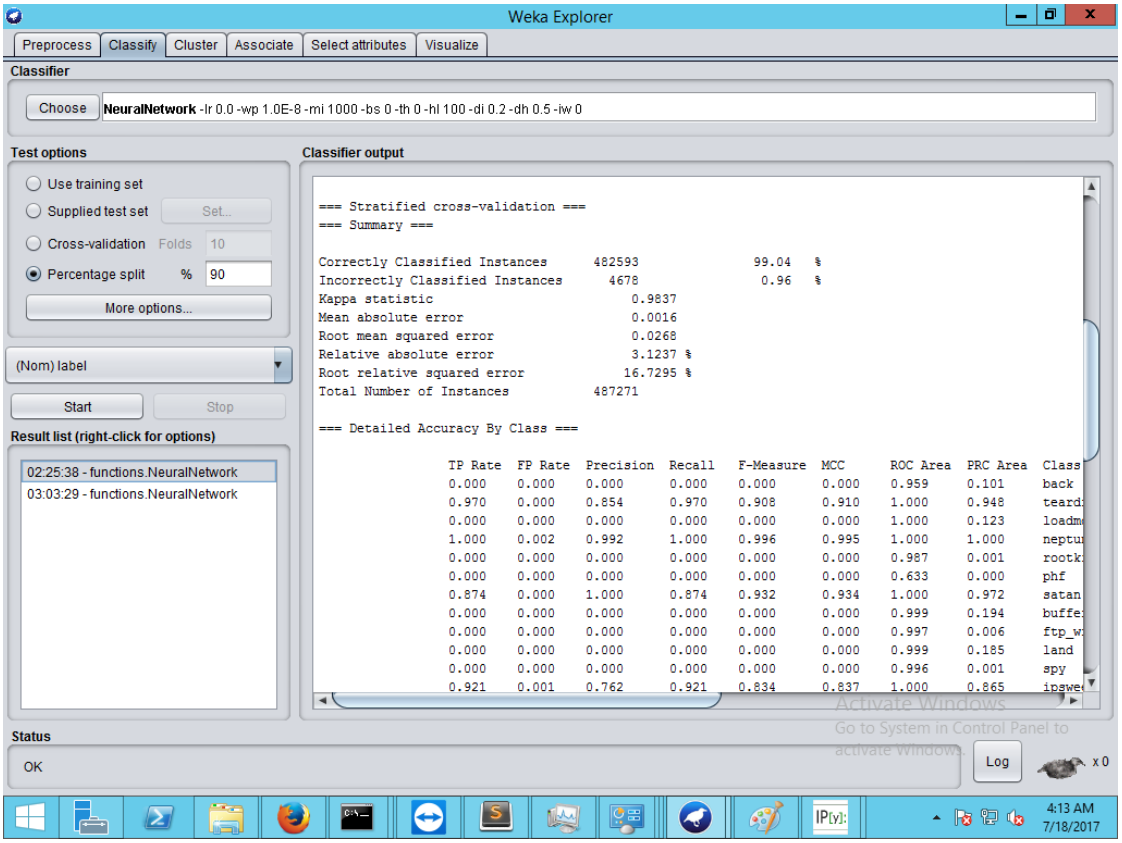

Figure 2: Detailed Accuracy by Class for Cross-Validation.

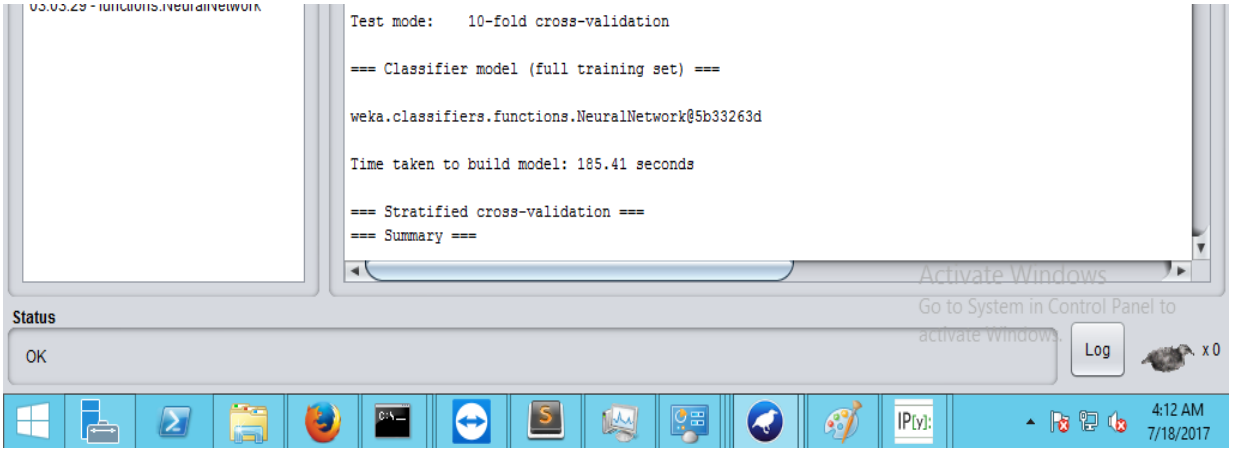

Figure 3: Total Time taken to build the model.

Table 5: Results Comparison with existing systems.

	Technique Used	Data Set	Accuracy
[18]	K-means +KNN+DT	KDDCup99	96.55%
	Genetic Algorithm + Support Vector Machine	KDDCup99	96.50%
	Support Vector Machine	KDDCup99	96.08%
	Support Vector Machine +BAT Algorithm	NL-KDD	97.38%
	Decision Tree	NSL-KDD	82%
	Cluster centre+ K- NN	KDDCup99	97.98%
Proposed System	CNN	KDDCup99	99.04%