Deep Learning Based Security Management of Information Systems: A Comparative Study

Cem B. Cebi, Fatma S. Bulut, Hazal Firat, and Ozgur Koray Sahingoz
Computer Engineering, Istanbul Kultur University, Istanbul, Turkey
Email: {cemberkecebi, fatmasenabulut, hazalfirat97, sahingoz}@gmail.com

Gozde Karatas
Mathematics and Computer Sciences, Istanbul Kultur University, Istanbul, Turkey
Email: g.karatas@iku.edu.tr

***Abstract*—In recent years, there is a growing trend of internetization, which is a relatively new word for our global economy that aims to connect each market sector (or even devices) by using the worldwide network architecture as the Internet. Although this connectivity enables excellent opportunities in the marketplace, it results in many security vulnerabilities for admins of the computer networks. Firewalls and Antivirus systems are preferred as the first line of defense mechanism; they are not sufficient to protect the systems from all types of attacks. Intrusion Detection Systems (IDSs), which can train themselves and improve their knowledge base, can be used as an extra line of the defense mechanism of the network. Due to its dynamic structure, IDSs are one of the most preferred solution models to protect the networks against attacks. Traditionally, standard machine learning methods are preferred for training the system. However, in recent years, there is a growing trend to transfer these standard machine learning based systems to the deep learning models. Therefore, in this paper, IDSs with four different deep learning models are proposed, and their performance is compared. The experimental results showed that proposed models result in very high and acceptable accuracy rates with KDD Cup 99 Dataset.**

***Index Terms*—cyber security, intrusion detection systems, deep learning, BiRNN, BiLSTM, CNN-LSTM, GRU, KDDCup99**

I. Introduction

In the era of technology, the utilization of networks and the number of devices connected to the Internet continuously increase. In order to accomplish organizational and personal actions in anytime and anywhere concept, an Internet connection is an inevitable requirement. Currently, people can share, store, and interact with data intensively by using this global network. With the growing need for the Internet, the issue of security for computer networks is a very trivial issue that needs to be overcome. While the new technologies and capabilities can close lots of security breaches, new types of threats, vulnerabilities, and attacks are encountered meanwhile.

Manuscript received February 16, 2020; revised July 3, 2020.

To protect the networks and assets of a company is the priority task of the security admins of the system. Any security breach or loss of data can cause drastic consequences. These consequences might lead to compromise of personal data, violation of laws and regulations, loss of money and also reputation. Therefore, the security of networks is one of the hot research areas in the world. On the other hand, securing the systems and data has become a challenging issue. Attacks are becoming more and more complex as methods and knowledge about previous attacks and vulnerabilities spread. Easy access to information makes individuals able to use different types of tools to exploit vulnerabilities without expertise. Therefore, it has become easier for attackers to intrude on systems.

Mainly firewalls are accepted as the first line of defense of the computer networks. Although it protects the system from outside attacks, it has a vulnerability about inside ones. Therefore, additional security mechanisms are also needed [1], [2]. In this point, Intrusion Detection Systems (IDSs) which try to detect the attacks not only from the outside of the company but also inside, are taken into consideration. In the literature, there are many different implementations of the IDS systems. According to their detection approach, IDSs can be classified in two different categories as Signature-based and Anomaly-based IDSs. In the former one, the system can detect intrusion according to the signature of the attack messages. Therefore, it needs to use a database that is required in order to be updated periodically. Although there is an excellent runtime efficiency for the detection process, these systems are vulnerable, especially zero-day attack which is not encountered previously.

To overcome this deficiency, anomaly-based systems, which firstly defines the regular message traffic and then identify the abnormal ones, are implemented. These systems mainly implemented with a learning mechanism to identify regular messages. Due to its dynamicity, Anomaly-based IDSs are mostly preferred. Generally, traditional machine learning approaches are used for training the system. However, with the improvement in computers' hardware and parallel implementation technologies, it is a relatively easy task to process big

doi: 10.12720/jait.11.3.135-142

data by using a new learning mechanism named Deep Learning.

In this paper, we have implemented Intrusion Detection Systems by using four different Deep Learning Approaches in a comparative study. We used a worldwide known and accessible dataset, KDDCup99, to test our proposed system. In the ongoing sections, we detailed the experimental results with the used parameters. These results showed that Deep Learning Methods could produce excellent accuracy rates, especially with the use of GPU architecture.

The rest of the paper is organized as follows: In the next section, some background knowledge in explained as Deep Learning and CUDA/GPU architecture. In Section III, related works on the topics are presented. The detailed of the prosed systems and Experimental results of the proposed methods are detailed in Section IV and Section V, respectively. Finally, conclusions and future works are drawn.

II. Background

A. Deep Learning

We can think of Deep learning as a more advanced version of the concept of machine learning. Deep learning is an approach used in the machine's perception and understanding of the world [3]. Deep learning architecture has been used in many subjects ranging from intrusion detection to Self-driving cars or cancer research without doctors. Targeted with deep learning is to prepare the necessary basis for the computer model to be able to do this editing instead of laying down a software step by step. In this way, the computer model in the face of alternative scenarios will be able to produce solutions through deep learning. While the scenarios that a programmer can create in the software stage of a system are limited; there are numerous solutions that deep learning structures can offer. Deep learning is an approach inspired by neural networks, a part of the human nervous system. In the deep network model, there are many interdependent artificial neurons in the complex order, and each neuron encodes a representation of the raw data at different abstraction levels within this structure consisting of multiple process layers.

B. Comparison of Machine Learning vs. Deep Learning

We use machine learning to parse data, learn from data, and make decisions based on what we have learned. Deep Learning is basically used to create an artificial neural network that can make intelligent decisions on its own. Deep learning is a subfield of machine learning. The main difference between the two algorithms is performance. When the number of data is low, deep learning algorithms do not perform well. Deep learning algorithms need large amounts of data to work well. In contrast, machine learning can give better results with fewer data.

To summarize for the problems involving big data, Deep Learning is the best solution, and for the issues with less data Machine Learning can give us the best solutions [4]. Also, the other notable difference between the two learning methods is time. In studies involving extensive data, machine learning is working more slowly, and deep learning gives much faster results. For these reasons, we chose to use a Deep Learning structure while developing our system. After the pre-processing of the dataset is completed, deep learning models are implemented to the selected dataset for intrusion detection. Four different deep learning architectures have been used for the research of the best accuracy that can be obtained with the KDD 99 dataset. The following are implementations of deep learning models: Deep Neural Networks, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network, Bidirectional RNN, Bidirectional-LSTM, Gated Recurrent Unit, Generative Adversarial Network

However, the primary advantage of the deep learning especially fits with big data concepts. Therefore, used models are directly related with data size and used application area.

C. Cuda

Cuda (Compute Unified Device Architecture), which is developed by Nvidia, is a parallel programming platform that makes a significant contribution to the computing performance of the computer. It can be defined as a system that allows written algorithms (programs) to work on the GPU (Graphics Processing Unit). It supports most of the popular programming languages such as C, C #, Java, and Python.

Since parallel programming requires more than one processor to work together, several problems arise. Complex software is necessary for many processors to be used together. CUDA eliminates these difficulties because it creates parallelism within it. Image processing, medical imaging, machine learning, mathematical operations, and many different processes can be done with a GPU used in CUDA [5].

III. Related Work

In this section, we first investigated the old studies on Intrusion detection on different electronic databases such as IEEE, Springer, and Research Gate. We selected about 20 articles that are relevant to our topic, and we did research on them. These articles we have worked on intrusion detection in different ways. Through these studies, we examined the existing types, techniques, and different architectures. In the final stage, we have continued our work in order to advance our project in a better direction by identifying the difficulties and problems in the current studies.

Karatas *et al.* explained the advantages and approaches to the use of DL for the solution of the IDS problem in [6]. In machine learning models the used parameters and also activation functions have important roles. In [7] authors investigate these parameters with different tests. Similarly, the effect of the used mechanism depending on the number of layers is discussed in [8] with a deep neural network approach. Yin *et al.* developed a model to detect four different types of attack these are User to Root

(U2R), Remote to Local (R2L), Denial of Service (DoS), and Probe [9]. The model works depending on the Recurrent Neural Networks. While developing the model, hyperparameters such as learning rate and neuron numbers were changed, and their effects on results were examined. Recurrent Neural Network model developed at the same time as compared with traditional machine learning algorithms. As a result of these comparisons, it was observed that the Recurrent Neural Network model was working with better accuracy than traditional machine learning algorithms.

Jing and Bin proposed a network intrusion detection algorithm using relevance deep learning [10]. This study emphasizes the importance of deep learning, which helps network intrusion detection. The intrusion detection algorithm proposed in this study consists of the data association stage and the deep learning stage. In the association stage, the network intrusion data is classified. This results in a reduction in the computational complexity of the algorithm by using encoding and decoding to set the relations between the high-speed ultra-high bandwidth network and space. The population optimization of detection data is achieved with crossover mutation and selection. Experimental results show that intrusion is detected 10 times compared to the detection of standard data, and the possibility to detect high-speed ultra-high bandwidth networks on average is higher than 50%. And also, the false detection rate is about 1.5%, which means relevance deep learning is an efficient algorithm to be used in network intrusion detection.

Niyaz and others represent a study that emphasizes deep learning approaches to help to overcome the challenges of developing an effective network intrusion detection system by collecting unlabeled data and using self-taught learning for unknown future attacks [11]. In this paper, in order to build an effective and flexible network intrusion detection system, an approach based on deep learning is proposed. For achieving this, a Sparse Auto-Encoder and Sort-Max Regression based network intrusion detection system was applied and evaluated with NSL-KDD. Results show that when assessed on the test data, the network intrusion detection system proposed in this study had good results compared to previously implemented ones in terms of the normal/anomaly detection. With the openness of the network, the existence of system security vulnerability, and the diversity of invasive technology, Traditional security technologies are not enough to detect intrusion.

Mirza and Cosan proposed a sequential Auto-Encoder framework using Deep Neural Networks (DNN) for network intrusion detection [12]. Due to the limited performance of DNN in modeling time series and processing temporal data, Recurrent Neural Networks (RNN) comes in the view, which copes with DNN's insufficient performance and handles the time dependencies. Auto-Encoders operate on the fixed length data sequence, and they detect subtle anomalies. Unlike regular Auto-Encoders, Recurrent Auto-Encoders can compress sequences with varying lengths. The proposed system uses sequential Long Short-Term Memory (LSTM) Auto-Encoders, which is a particular case of RNN. An online sequential unsupervised framework for network intrusion detection using LSTM Auto-Encoders is developed, which is dynamic and scalable since it can work on both fixed and variable length data. And experiments show that the proposed algorithm achieves a best harmonic average of the precision and recall (f1 score) and Area under the Curve (AUC) score, also compared with other proposed algorithms, the best f1 score is achieved with the LSTM Auto-Encoder with max pooling and Deep Auto LSTM Networks.

Zhao and others proposed an intrusion detection method based on Deep Belief Network (DBN) and Probabilistic Neural Network (PNN) [13]. This method is proposed to solve some problems with intrusion detection. These problems are easy to fall into the local optimal, a large amount of data, redundant information, long-time training. Different deep learning methods can be used to overcome some of these problems. Deep learning provides a new way for a machine to make statements in each layer with the information gained from the previous layer by using Back Propagation (BP) algorithm. Since deep learning is convenient to detect various high dimension invasion behavior, this study makes use of deep learning. First, in order to bring out the features of the data, the raw data is processed to be low-dimensional data by using the nonlinear learning ability of Deep Belief Network (DBN). Second, the Probabilistic Neural Network (PNN) is used for the intrusion detection model. Next, Particle Swarm Optimization (PSO) algorithm is used for the number of nodes in the hidden layer. And finally, KDD CUP 1999 dataset is used for testing and training of the model. Experimental results show that DBN, PSO algorithm, and PNN combined are useful and helps to find solutions for the mentioned problems.

Zhao *et al.* proposed a new intrusion detection system based on a neural network that provides self-organization, self-learning, pattern recognition, generalization to speed up detection, increase the accuracy and provide sufficient protection for internal attacks, external attacks and misuse with combining of the expert system detection and training data set. Thanks to the neural network training set, several of attack type is detected and increase the detection accuracy compared to traditional security technologies [14].

Lin *et al.* present a research paper about network intrusion detection using Convolutional Neural Networks (CNN) based on LeNet-5 to detect threats based on network and KDD Cup 1999 data set is used [15]. According to test results, the prediction accuracy increases to 99.65% with more than 10,000 data. The total accuracy rate is 97.53%. To increase the accuracy rate of threat detection, this study provides an improved behavior-based classifier learning model for anomaly detection based on Convolutional Neural Network

As a result of our research, most of the currently used Intrusion detection systems suffer from the following problems in general.

1) Intrusion detection systems use system resources continuously. The system monitors even if there is

no intrusion because the components of the Intrusion detection system should always be active. This is a resource usage problem.

2) The data used by Intrusion detection systems are obtained from packages in the network. The data must pass a long path from the mainstream to the Intrusion Detection System, and the packet can be changed or destroyed by a potential attacker.
3) An intrusion detection system is applied as a separate program, and therefore, it is sensitive to external interference. The potential attacker can interfere or disable the program which runs in the system. This is a problem of reliability.
4) Overfitting problems are encountered in developed Deep Learning models. This occurs because the model memorizes the training data set and causes this model to fail against the newly acquired data (low accuracy).
5) Some developed Deep Learning models have time problems. This is a massive problem for intrusion detection systems. Because the incoming attack should be identified as quickly as possible, and precautions should be taken against it.

IV. Proposed System

In this study, we propose an intrusion detection system with deep learning by using the KDD CUP 99 dataset. The main idea of using deep learning for intrusion detection system is to achieve an effective system for preventing attacks with signatures similar to anomalies without the need for human expertise set of rules to follow.

A. Dataset

KDD CUP 99 dataset is one of the most widely used datasets for intrusion detection systems and derived from DARPA 98 intrusion detection evaluation dataset. KDD CUP 99 dataset contains malicious activities simulated in a military network environment. Malicious activities included in the dataset are classified into four categories, which are Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), Probing Attack. KDD CUP 99 dataset features can be examined under 2 categories: Basic Features Traffic Features "Same Host" Features and "Same Service" Features Since KDD CUP 99 is derived from DARPA 98, issues existing in this dataset is also present in KDD CUP 99, and there are also other problems discussed regarding KDD CUP 99 dataset some of which are irregularities, paper content being out of date, meaningless, false positive results and redundant records [16].

B. Data Preprocessing

Data pre-processing was performed to handle the KDD CUP 99 dataset to be compatible with deep learning models. In the dataset 'protocol_type,' 'service', 'flag' and 'label' are not in the numerical form. These values were digitized during the pre-processing phase. The 'protocal_type' property has 3 different values in the dataset, namely 'icmp', 'tcp' and 'udp', which were digitized 0, 1, 2 respectively. In the dataset 'service' feature is 66, and the flag feature has 11 different values. The values of these properties were numbered, starting from 0 up to the amount they were found. Finally, there are a total of 23 values for the 'label'; attack classifications are digitized from 0 to 22.

C. Programming Environment

For the implementation of the intrusion detection system, various tools and technologies have been used. Python programming language is used for both dataset pre-processing and building of deep learning models for intrusion detection because of the rich selection of libraries available. Python-specific Integrated Development Environments (IDE) used during the study are PyCharm and Spyder IDE. For the execution of developed codes on GPU using CUDA, Python 3.6 Shell and Command Prompt are also used.

As stated, Python programming language has a wide range of libraries for machine learning, deep learning, computation, and data structures. Tensorflow, Keras libraries are used to build and train deep learning models. We mainly use Keras because Keras is user-friendly, modular, easy to expand, and most importantly, created to work with python. Neural layers, optimizers, cost functions, regularization schemas are independent modules that we use to develop new models [17]. Models are defined directly in python code; there is no separate model configuration.

The primary reasons why we prefer Keras are primarily due to the principles of user-friendliness, multi-explanatory, and has documentation of very high standards. Beyond the ease of learning and the ease of model creation, Keras has a wide range of applications, integration with five different back-end engines (Theano, Tensorflow, CNTK, PlaidML, and MXNet) and strong support for multiple GPUs and distributed training. It is also supported by companies like Amazon, Apple, Nvidia, Uber, and Google. In addition, libraries such as NumPy, pandas, scikit-learn, and Matplotlib are utilized. memory.

For the proposed system, CNN-LSTM, Bidirectional RNN, Bidirectional LSTM, and GRU deep learning hybrid architectures have experimented on the dataset, and we aimed to create a successful intrusion detection system by using hybrid structures.

D. Deep Learning

The primary purpose of this part is giving a general overview of Deep Learning. Firstly, this part provides general information about the Bidirectional Recurrent Neural Network (BiRNN), Bidirectional Long-Short-Term Memory network (BiLSTM), convolutional - Long - Short - Term memory network (CNN-LSTM) and Gated Recurrent Unit (GRU).

BiRNN: In order to fully understand the structure of the Bidirectional Recurrent Neural Network (BiRNN), we will first look at the Recurrent Neural Network (RNN) structure. RNN is used to understand the structure of the input sets that come with a particular order in a time-dependent situation and to produce the output [18]. The data that is used as the output in the previous process is

used as the input of the next process. A recurrent neural network is different from deep neural network techniques based on a feed-forward neural network. Because the data in the feed-forward neural network are processed forward and the data is not returned. The recurrent neural network must contain a memory in order to provide dependency between outputs in previous processes and input of the next process.

Although RNN is thought that long-term dependencies are achieved theoretically, practically RNN presents limited achievement. Another drawback of RNN is the Vanishing Gradient problem. Since all layers and time-dependent steps have connected each other by multiplying, their derivatives are faced with the danger of extinction or rise. This is called Vanishing Gradient Problem

Bidirectional recurrent neural networks connect two hidden layers running with opposite directions to one output, permitting them to receive data from each past and future situation. This technique is mostly used in supervised learning because it is difficult to calculate reliable probabilistic models in unsupervised learning models [19].

BiLSTM: LSTM that is a new type of RNN, was developed in order to solve Vanishing-Gradient problem and deal with long-term dependencies due to short-term memory. Unlike the RNN, LSTM contains cells that are called memory. There are also gates that are connected to each other in a particular way within the LSTM. These gates are known as Input Gate, Forget Gate, and Output Gate that result in a value between 0 and 1. According to the result, it becomes clear that what will happen to the data, whether the data will forget, select, or collect. The data sets that come in a time-dependent situation combined with the memory that contains the previous outputs and produces new output that is stored into the memory [20], [21].

Bidirectional LSTM runs your inputs in 2 ways that, from past to future and from future to past. With this approach, bidirectional LSTM runs backward and preserve the information from the future. Also, Bidirectional LSTM using two hidden states that combined with each other, with the combined states, the model will able to preserve information from the past and the future.

CNN-LSTM: We have examined the LSTM architecture in the CNN-LSTM hybrid structure, and we will first investigate the CNN structure and then the hybrid structure. Convolutional neural networks (also known as CNN or CovNet) is a concept that has emerged by modifying traditional neural networks. Such systems are a kind of deep or profound learning neural networks due to the wide and deep structure [22]. Convolutional neural networks are very popular nowadays due to their great success in the classification of picture-based objects.

In an image processing with CNN, all pixels must be transferred to the neural network to classify or recognize a picture with traditional neural networks. In convolutional networks, at first, some patterns are tried to be detected on the picture, and then these patterns are transferred to the neural network. This way model can achieve more successful results with fewer complex formations. For example, when we want to categorize the pictures as cat or non-cat. In general, the ear, mouth, and tail shapes of the cats are extracted from known cat pictures then these shapes are searched as if they exist in the new images. In convolutional networks, these shapes are called filters. Thus, we will classify the images that contain these patterns as cats and those that do not. Convolution is the first layer for the extract features from the input; Convolution detects the relationship between pixels by learning small parts of the input. Also, the Pooling layers reduce the size of the parameters when the inputs are too large. In the Fully Connected part, we flatten our input matrix into a vector. Then we feed our vector into a fully connected layer as a neural network [23].

CNN-LSTM neural networks are composed of CNN and LSTM layers. CNN-LSTM architecture makes use of CNN layers for feature extraction in order to select useful features on input data, and LSTM layers are used for their ability to cope with sequential data. This architecture is specially designed for the need for predicting sequence problems with spatial inputs like images or videos. CNN-LSTM architecture can be implemented for various issues like speech recognition, image processing, natural language processing, and language translation

GRU: Gated Recurrent Unit was developed to solve the vanishing gradient problem in RNN in 2014. GRU is a variance of LSTM in terms of the design and produced accuracy. In some cases, GRU is separated from LSTM. For example, the LSTM has three gates, while GRU has two gates: the reset gate and the update gate. The Update gate acts as a forget and input gate in the LSTM model. Decides which information is to be kept and which data is to be thrown away. The reset gate is another gate for determining how much of the past data is forgot [24].

V. Experimental Results

The execution of deep learning models is performed using a single computer. Operations with datasets are executed on a computer with specifications stated in Table I. We have used KDD'99 multiclass dataset for obtaining the result of accuracy and learning time in Table II. Models were trained and then tested by four different deep learning architectures mentioned in this study with the train and test dataset that are part of the dataset with 494,021 samples and split with the percentage of 80 and 20 respectively which corresponds to 395,217 samples for a train set and 98,804 samples for the test set. Bidirectional RNN, Bidirectional LSTM, CNN-LSTM, and GRU models have 2 layers. We chose batch size as 1,000 and epoch as 100 for all the models of deep learning architectures. These models are executed on GPU.

The architecture with the best accuracy was Bidirectional LSTM with a value of 0.9993, as can be seen from Table II. Accuracy and loss graphs of models can be seen in Fig. 1-Fig. 4.

TABLE I. COMPUTER SPECIFICATIONS

Component	Component Name
Processor	Intel(R) Core (TM) i7-8700 CPU @ 3.20GHz
RAM	8 GB RAM
Graphics Card	NVIDIA GeForce GTX 1080 Ti 11 GB GDDR5X, 11GHz, 1582 MHz, 3584 CUDA Cores
Operation System	Windows 10 Pro 64 bit

TABLE II. ACCURACY AND LEARNING TIME OF THE MODELS

Method	Accuracy (%)	Time (sec)
BiRNN	99.92	244.09
BiLSTM	99.93	529.10
GRU	99.71	215.81
CNN-LSTM	99.87	1127.11

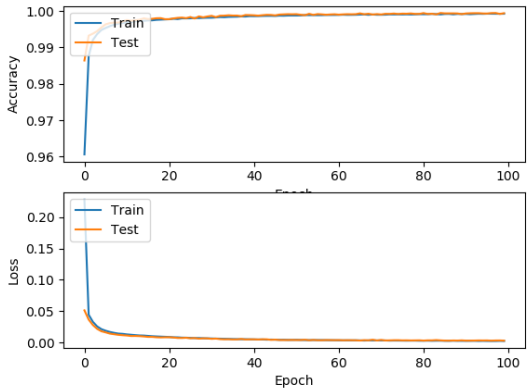


Figure 1. Train and test for bidirectional RNN.

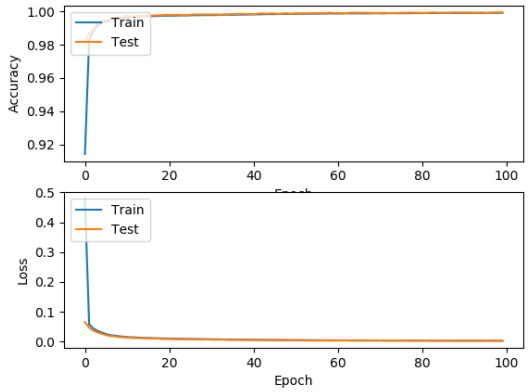


Figure 2. Train and test for bidirectional LSTM.

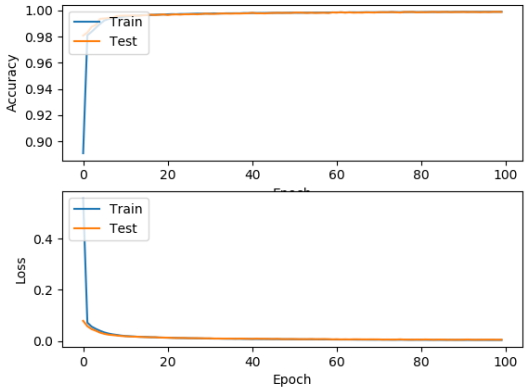


Figure 3. Train and test for GRU.

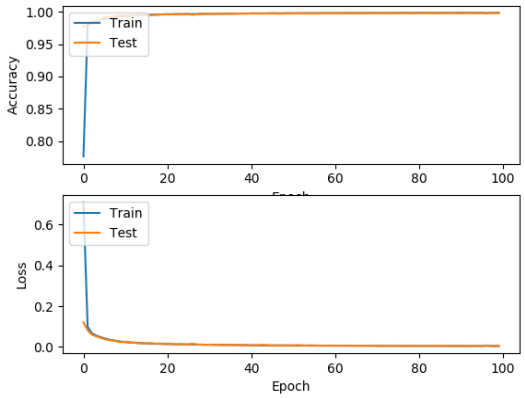


Figure 4. Train and test for CNN-LSTM.

In Bidirectional Recurrent Neural Network, we use two Bidirectional layers and one output dense layer. Inside the first Bidirectional layer, we create a Simple Recurrent Neural Network (RNN) with 128 nodes; for the activation layer, we use Rectified Linear Units (RELU) and the finally we give return sequences parameter a true value. Also, for the Bidirectional layer, we assign the input shape parameter to (1, 82). After the first hidden Bidirectional layer, we create a dropout with a rate of 0.3. In the second hidden Bidirectional layer, we create the layer with 64 nodes, for the activation layer, we use RELU and return sequences parameter is set to false. After that, we add an output dense with 10 node size, and for the output node, we use the Softmax activation function.

In the Bidirectional Long Short-Term Memory (LSTM) we use two hidden Bidirectional layers and one output dense layer. For the first Bidirectional layer, we create an LSTM layer with 128 nodes; for the activation layer, we use RELU, return sequences parameter is set to True, and recurrent_dropout is set to 0.5. Also, for the Bidirectional layer, we assign the input shape parameter to (1, 82). After the first layer, we add a dropout with a rate of 0.3. Secondly, we create a second hidden Bidirectional RNN layer with the 64 nodes; for the activation layer, we use RELU, return sequences parameter is set to False this time, and recurrent_dropout is set to 0.5. Finally, we add an output dense with 10 node size, and for the output node, we use the Softmax activation function.

We aimed to design a GRU model with sequential layers. We've added two fully connected hidden layers, where all inputs and outputs are connected to each other and one output dense layer.

The first layer's input corresponds to the features list number of the data set. The number of units of the first two hidden layers that correspond to the number of output is 128, 64 respectively, and the RELU activation function for these layers. In order to prevent the overfitting problem, one dropout function is added at a rate of 0.3 in the between the first two layers. The return sequence value of the first layer is set. True and second layer's Return_sequences value is false. The output dense layer's number of units is 10. because the number of outputs of the data set is 10. The activation function of the output layer is Softmax.

After the necessary functions have been defined, and the hyperparameters are set, the GRU model is compiled. Two parameters are used for compile function: Optimizer and Loss. We chose categorical_crossentropy for the loss function, Adam algorithm for the optimizer. Then the model is given data to train. This training is performed by renewing the data set 100 times and running the optimization algorithm in the form of 1,000 samples of mini-batches. At the same time, the GRU model is trained by using 80 percent of the dataset and test it with the remaining 20 percent.

During the implementation of Convolutional Neural Network - Long-Short Term Memory (CNN-LSTM), a sequential model with one 1D Convolution layer, one Max Pooling layer and one LSTM layer is used. As an input to the first layer, which is one dimentional Convolution layer, input length with the number of features in the dataset is passed, which is 82. The first layer contains 128 filters with kernel size 3 and activation function set as RELU. The first layer, which is the Max Pooling layer that reduces the dimensions of the data, has pool length 2. The second layer, which is the LSTM layer, contains 64 nodes. Between the first layer and second layer, dropout with rate 0.3 is applied to the input to drop out units for preventing overfitting. A number of nodes in the output layer of the model are equal to the number of classes that can be obtained in the dataset, which is 10. The activation function of the output layer is set as Softmax.

After the model is created, configurations for the learning process have been arranged. The optimizer is set as Adam and the loss function is set as Categorical Cross-Entropy. After the configuration process, the model is trained by iterating on the data in batches of 1,000 samples with a number of epochs set as 100. The model is trained with 80% of the dataset as training data, and the rest is used for the test dataset.

The run time efficiency of the proposed system is obtained by sending 15, 50, 100 and 1000 sample data to the learned models in order to analyze the time. to estimate the incoming attack classes of the deep learning models are shown in Table III.

TABLE III. RUNTIME EFFICIENCY (FOR CHECKING PACKETS)

Model	Time (milliseconds)			
	15 samples	50 samples	100 samples	1,000 samples
BiRNN	2	4	7	30
BiLSTM	3	5	11	82
CNN-LSTM	9	18	35	282
GRU	1	2	4	16

VI. CONCLUSION AND FUTURE WORK

In this article, we have introduced and implemented an Intrusion Detection System using different deep learning algorithms. The primary purpose of the system is to effectively detect network attacks by using deep learning. We used the KDD CUP 99 dataset to measure and implement the performance of the system and achieved a reasonable detection rate. Firstly, we took a part of our system and ran it on four different deep learning models. At the same time, we have examined other studies on parameters for our algorithms and applied them to our system. After running our models on test datasets, we used the best results for our original datasets and made the necessary measurements.

In the future, we are going to apply our models to real systems. Also, we will make practical analyses and mitigate the problems that can be faced. Additionally, some new approaches like use of clustering can be applicable to reach the better results [25].

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Sahingoz and Karatas conducted the research and proposed the algorithm; Firat prepared the dataset; Cebi programmed for the model; Bulut did the experiment; all authors had approved the final version.

ACKNOWLEDGMENT

Participation in this Conference was supported by Istanbul Kultur University [TPDB 2019/2020-61].

REFERENCES

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

[2] Y. Jia, M. Wang, and Y. Wang, "Network intrusion detection algorithm based on deep neural network," *IET Information Security*, vol. 13, no. 1, pp. 48-53, 2019.

[3] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *Proc. 31st Youth Academic Annual Conference of Chinese Association of Automation*, 2016.

[4] Y. Xin, L. Kong, Z. Liu, and Y. Chen, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365-35381, 2018.

[5] What is CUDA. The Official NVIDIA Blog. [Online]. Available: https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/

[6] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," in *Proc. International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism*, 2018.

[7] G. Karatas and O. K. Sahingoz, "Neural network based intrusion detection systems with different training functions," in *Proc. 6th International Symposium on Digital Forensic and Security*, Antalya, 2018, pp. 1-6.

[8] B. Reis, S. B. Kaya, G. Karatas, and O. K. Sahingoz, "Intrusion detection systems with GPU-accelerated deep neural networks and effect of the depth," in *Proc. 6th International Conference on Control Engineering & Information Technology*, Istanbul, Turkey, 2018, pp. 1-8.

[9] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017.

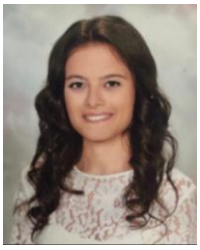
[10] L. Jing and W. Bin, "Network intrusion detection method based on relevance deep learning," in *Proc. International Conference on Intelligent Transportation, Big Data & Smart City*, 2016.

[11] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proc. the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016.

[12] A. H. Mirza and S. Cosan, "Computer network intrusion detection using sequential LSTM neural networks autoencoders," in *Proc.*

26th Signal Processing and Communications Applications Conference, 2018, pp. 1-4.
[13] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *Proc. IEEE International Conference on Computational Science and Engineering*, 2017, vol. 1, pp. 639-642.
[14] J. Zhao, M. Chen, and Q. Luo, "Research of intrusion detection system based on neural networks," in *Proc. IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 174-178.
[15] W. H. Lin, H. C. Lin, P. Wang, B. H. Wu, and J. Y. Tsai, "Using convolutional neural networks to network intrusion detection for cyber threats," in *Proc. IEEE International Conference on Applied System Invention*, 2018, pp. 1107-1110.
[16] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
[17] Keras: The Python deep learning library. *Keras Documentation*. [Online]. Available: https://keras.io
[18] Build with AI. DeepAI. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/bidirectional-recurrent-neural-networks
[19] H. Ergüder. (2018). Recurrent neural network Nedir? [Online]. Available: https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120
[20] N. Donges. Recurrent neural networks and LSTM. *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5
[21] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. International Conference on Platform Technology and Service*, 2016.
[22] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. International Conference on Advances in Computing, Communications and Informatics*, 2017.
[23] M. Sewak, R. Karim, and P. Pujari, *Practical Convolutional Neural Network Models*, Packt Publishing Ltd., 2018.
[24] R. Dey and F. M. Salemt, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," in *Proc. IEEE 60th International Midwest Symposium on Circuits and Systems*, 2017.
[25] B. Reis, S. B. Kaya, and O. K. Sahingoz, "A clustering approach for intrusion detection with big data processing on parallel computing platform," *Balkan Journal of Electrical and Computer Engineering*, vol. 7, pp. 286-293, 2019.

Cem B. Cebi was born in Istanbul, TURKEY in 1997. He received the Bachelor of Science degree from the Computer Engineering Department of Istanbul KulturUniversity in 2019. After graduation from Istanbul Kultur University, he joined the engineering team at Iyzico where he works on a wide variety of exciting and meaningful projects.



Fatma S. Bulut was born in Istanbul, TURKEY in 1996. She graduated third out of computer engineering department and received the B.S degree from the Computer Engineering Department of Istanbul KulturUniversity in 2019. She started to work as a software engineer at ING Bank in 2019. She aims to learn new technologies related to her field.

Hazal Firat was born in 1997. She graduated second out of Computer Engineering Department and third out of Engineering Faculty and received a B.S. degree from Computer Engineering Department of Istanbul Kultur University in 2019. She worked as a security trainee at Sony for 6 months during her final-year of the university. She is currently looking for new opportunities in her related field.

Gozde Karatas was born in Istanbul, TURKEY in 1991. She received her undergraduate degree from Mathematics and Computer Science Department of Istanbul Kultur University in 2009 and her graduate degree from Computer Engineering Department of Istanbul Kultur University in 2013. In 2015 she completed her master thesis on NoSql Database Testing in Istanbul Kultur University. During her master studies, she worked on distributed databases. Also, she has been working at the Department of Mathematics and Computer Science in Istanbul Kultur University as Research Assistant. She is currently a PhD candidate at Marmara University in Computer Engineering Department and continues to work in the field of computer security. Her research interests include computer networks and security, machine learning, deep learning, cryptography, python programming, statistics and graph theory.

Ozgur K. Sahingoz received the B.S. degree from the Computer Engineering Department of Bogazici University in 1993 and M.S. and PhD degrees in Computer Engineering Department of Istanbul Technical University in 1998 and 2006 respectively. He is currently working as an Associate Professor in the Computer Engineering Department of Istanbul Kultur University. He is the author of more than 100 papers, and he is still working in two research projects. His research interests include artificial intelligence, machine/deep learning, data science, software engineering, and UAV Networking.
He graduated more than 13 MSc students and supervised around 6 Ph.D. students. He has reviewed more than 80 national projects especially related to TUBITAK, KOSGEB-Ministry of Industry and Technology (Turkey). He is also a regular reviewer for more than 40 Science Citation Index (/Expanded) international journals.
He has also been very active in scientific conferences, organized and/or works as program committee member more than 100 conferences/workshops on different research areas, especially on artificial intelligence and information sciences. He has developed and taught around 20 various academic courses.