IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

Botnet Detection in IoT

Manoj H V, M S Omprakah, Manogna K M, PavanA M, Prof. Ashok K N
Department of Computer Science and Engineering
SJC Institute of Technology, Chikkaballapur, Karnataka, India

Abstract: *The proliferation of IoT devices has brought about a higher risk of botnet attacks, which can cause severe damage to networks and devices. A botnet refers to a network of compromised devices that are exploited to carry out various malicious activities, including DDoS attacks, spamming, and data theft. Given the significance of IoT device security, it is crucial to develop effective methods for detecting botnets. To address this issue, we propose a hybrid deep learning technique for botnet detection. Our approach revolves around configuring network traffic routing, which allows us to identify botnet attacks by analyzing patterns in network traffic. Deep learning models, such as Long Short-Term Memory (LSTM) and Deep Neural Networks (DNN), play a crucial role in enhancing the accuracy of detection. These models enable the system to learn from historical patterns and adapt to new ones, improving its ability to identify and flag botnet activity.*

Keywords: IoT, Deep learning, LSTM, DNN, DDoS, security

I. INTRODUCTION

The rapid expansion of IoT technology across various sectors such as surveillance, healthcare, transportation, manufacturing, and education highlights the need to prioritize the security of IoT infrastructure to enhance its performance. To address this, existing systems extensively examine the threats, security requirements, challenges, and attack vectors that are specific to IoT networks. By conducting a thorough gap analysis, attention is drawn towards a network-based deployment of IoT architecture.

In this context, a promising approach for strengthening the security of IoT networks is the utilization of deep learning (DL) in fog computing applications. Fog computing refers to a decentralized computing infrastructure that extends cloud computing capabilities to the edge of the network. DL, a subset of artificial intelligence, can play a vital role in detecting and countering security threats in fog computing systems. By leveraging DL algorithms, the system can adapt to different conditions and effectively identify abnormal behaviours within the network.

The proposed DL-based detection approach holds the potential to significantly enhance the overall performance of IoT systems. By continuously analyzing network behaviour, the system can recognize patterns associated with malicious activities or deviations from normal operation. This proactive approach allows for timely detection and response to potential threats, thereby improving the security and performance of the IoT infrastructure.

II. LITEREATURE REVIEW

Security is a crucial area of research in various networking paradigms such as cloud computing, fog computing, IoT, and SCADA systems. One specific focus of researchers is on detecting botnet attacks, which involve identifying infected devices before they can carry out malicious activities on the network. To address this challenge, numerous methods have been proposed, particularly those utilizing artificial intelligence, specifically machine learning (ML) and deep learning (DL) algorithms.

Different ML techniques have been employed for botnet detection, including BayesNet, Support Vector Machine (SVM), J48, Decision Tree (DT), and Naive Bayes (NB). ML methods can be categorized as supervised, unsupervised, or semi-supervised learning. For instance, Parakash et al. conducted experiments using three well-known ML algorithms, namely K-Nearest Neighbors (KNN), SVM, and NB, to detect DDoS packets. Their findings showed that KNN outperformed the other algorithms with 97% accuracy, while SVM and NB achieved 82% and 83% accuracy, respectively.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

In another study, authors proposed a detection scheme using the SVM algorithm along with their own proposed idle timeout adjustment algorithm (IA). They demonstrated that their methodology outperformed other approaches and achieved better results. Similarly, in a different work, the SVM algorithm, along with neural networks (NN) and NB models, was used for botnet detection. The NN and NB models achieved 100% accuracy, while the SVM model achieved 95% accuracy. Another study by Yeetal utilized the SVM algorithm and achieved an average accuracy of 95.24%

Additionally, authors in a different work achieved a 99.6% detection accuracy rate using Naive Bayesian and decision tree classifier algorithms.

While ML algorithms have their advantages, they face challenges in handling scalability, learning from massive data, and low-value density data. To address these challenges, DL algorithms, a subset of ML, have emerged as a preferable choice for IoT applications due to their ability to handle large datasets and unstructured data. DL and hybrid DL approaches have been applied for detecting various kinds of malware in IoT devices.

For instance, a study described a technique for defending the IoT environment against malware and cyber-attacks using DL in Software-Defined Networking (SDN). The proposed model achieved 99.87% accuracy, 0.0554% false positive rate, and had a low testing time. Another study implemented DL using two-way Long Short-Term Memory (LSTM) for packet-level inspection on IoT and networks, achieving satisfactory results. SDN was also utilized in another work to deploy a detection mechanism system for safeguarding IoT, achieving 95% accuracy in attack detection using the Restricted Boltzmann Machine (RBM).

The research in the field of network security has focused on detecting botnet attacks using various ML and DL algorithms. ML algorithms such as KNN, SVM, NB, and hybrid models have been employed for botnet detection, while DL algorithms have shown promise in handling extensive and unstructured data generated by IoT devices. The proposed approaches have demonstrated high detection rates and improved accuracy, contributing to the development of effective security measures in networking paradigms.

III. OBJECTIVES OF THE PROPOSED APPROACH

The most important objective of the proposed system is providing the security for the IoT devices. The main aim is to develop a flexible, reliable, and highly accurate system for detecting benign and malicious behavior. Monitoring the network traffic plays a crucial role in providing the security. Another important objective is to develop a model which can predict the botnet attacks with high accuracy and at the same time providing an easy software interface which can be used by normal user. The prediction algorithm will assist in identifying patterns of network based on specific fields and will orderly validate the relevant fields.

Additionally, the strategy offers a successful, adaptable method for identifying botnet attacks performed on IoT devices.

IV. METHODOLOGY

The model uses Deep learning algorithm for botnet detection in IoT. The system uses these DL methods to detect benign and malicious behavior in network. The whole dataset is divided into train and test datasets. We mainly use DNN and LSTM algorithms for detecting botnet attacks.

A. Deep Neural Network (DNN):

The DNN (Deep Neural Network) algorithm is a type of artificial neural network that consists of multiple layers of interconnected nodes (neurons). Each node receives input from the nodes in the previous layer, performs a computation, and passes the output to the nodes in the next layer. DNNs are widely used in various machine learning tasks, including botnet detection in IoT networks. The working of DNN model is as follows:

1. Model Initialization:

The DNN model is initialized using the Sequential class, which allows for building a sequential stack of layers.

2. Input Layer and Hidden Layers:

The first step is to add an input layer to the model. The input layer receives the input data, which in this case would be the preprocessed features from the IoT network traffic. Each hidden layer is then added to the model using the Dense

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

class. The Dense layer is a fully connected layer where each neuron is connected to every neuron in the previous layer. The number of hidden layers and the number of neurons in each layer are hyperparameters that need to be determined based on experimentation and optimization.

3. Activation Function:
An activation function is applied to the output of each neuron in the hidden layers. The activation function introduces non-linearity into the model, allowing it to learn complex relationships in the data. The most commonly used activation function for hidden layers is the rectified linear unit (ReLU), which maps negative inputs to zero and keeps positive inputs unchanged.

4. Output Layer:
The final layer of the DNN is the output layer, which is responsible for producing the predictions for the different classes (benign, botnet types). The number of neurons in the output layer corresponds to the number of classes to be classified. For multi-class classification, the softmax activation function is typically used in the output layer. Softmax normalizes the outputs into probabilities, ensuring that the predicted probabilities for all classes sum up to 1.

5. Model Compilation:
Before training the DNN model, it needs to be compiled with the appropriate optimizer, loss function, and evaluation metrics. The optimizer, such as Adam, determines how the model's weights are updated during training to minimize the loss function. The loss function, such as categorical cross-entropy, measures the difference between the predicted probabilities and the actual labels. Metrics like accuracy can be specified to evaluate the performance of the model during training and validation.

6. Model Training:
The model is trained using the fit method, which takes the training dataset (features and corresponding labels), as well as other parameters like the number of epochs (iterations over the training data) and the batch size (number of samples processed before updating the weights). During training, the DNN adjusts its internal weights based on the computed gradients and the selected optimizer to minimize the loss function.

7. Model Evaluation:
After training, the model can be evaluated on a separate testing dataset to assess its performance. The evaluate method computes the loss and accuracy (or other specified metrics) on the testing dataset.
The DNN algorithm learns to identify patterns and relationships within the input data through the training process. By adjusting the weights and biases of the neurons in each layer, the DNN can capture complex representations and make predictions on new, unseen data.

B. Long Short-Term Memory (LSTM):
Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is particularly effective in capturing long-term dependencies and patterns in sequential data. LSTM has been widely used in botnet detection in IoT networks due to its ability to handle time-series data and retain information over longer sequences. Here's a detailed explanation of LSTM and its application in botnet detection in IoT:

1. Recurrent Neural Networks (RNNs):
RNNs are designed to process sequential data by introducing feedback connections that allow information to persist over time. Traditional RNNs suffer from the vanishing gradient problem, where the gradients become too small as they propagate back through time, making it difficult for the model to learn long-term dependencies.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

2. LSTM Architecture:

LSTM was introduced to address the limitations of traditional RNNs in capturing long-term dependencies. The core idea of LSTM is the inclusion of a memory cell that can store and retrieve information over long sequences. The memory cell is composed of three main components: an input gate, a forget gate, and an output gate. These gates regulate the flow of information into, out of, and within the memory cell, enabling the model to selectively retain or discard information.

3. Input Gate:

The input gate determines which parts of the input should be stored in the memory cell. It takes the input at the current time step and the output from the previous time step and applies a sigmoid activation function. The resulting values indicate how much of the input and previous output should be stored in the memory cell.

4. Forget Gate:

The forget gate controls which information in the memory cell should be discarded. It takes the input at the current time step and the output from the previous time step and applies a sigmoid activation function. The resulting values determine how much of the information in the memory cell should be forgotten.

5. Memory Cell Update:

The memory cell is updated based on the input at the current time step and the output from the previous time step. The input is passed through a tanh activation function to generate a new candidate value for the memory cell. The input gate and the candidate value are multiplied together to determine which parts of the candidate value should be stored in the memory cell.

6. Output Gate:

The output gate controls the information that is retrieved from the memory cell and outputted to the next layer or as the final prediction. It takes the input at the current time step and the output from the previous time step and applies a sigmoid activation function. The resulting values determine how much of the information in the memory cell should be outputted.

7. LSTM in Botnet Detection:

In the context of botnet detection in IoT networks, LSTM is applied to analyze time-series data representing IoT network traffic. Each time step corresponds to a specific moment in time, and the input at each time step consists of features derived from the network traffic. The LSTM model processes the sequence of inputs, captures temporal dependencies, and learns patterns indicative of botnet behaviors. The output of the LSTM model can be used for binary classification (botnet or non-botnet) or multi-class classification (different types of botnets).

By leveraging its memory cell and gate mechanisms, LSTM can effectively capture long-term dependencies and temporal patterns in IoT network traffic data. This makes it a valuable tool in botnet detection, as it can model the dynamic behavior of network traffic over time and identify subtle indicators of botnet activity. The LSTM model is trained on labeled data and can be evaluated based on various metrics to assess its performance in detecting botnets in IoT networks.

The working of various phases in detection of botnet is as follows:

1) Dataset preparation:

Collecting the dataset that includes IoT network traffic data and labeling it to indicate whether each sample is benign or part of botnet. Splitting the dataset into train and test dataset.

2) Pre-Processing:

The dataset is pre-processed in order to improve overall performance of system. We first inspected dataset and removed NaN, infinite or missing values. Using MinMaxScaler, we normalized the dataset to a standard range. In pre-processing stage, the One Hot Encoding is performed to normalizing the data according to number of categories.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

3) Model Architecture Design:
The number of neurons, layers, sequence and other hyperparameters are determined in the design phase. The Softmax function is used to perform classification of attacks in an highly accurate manner. This Softmax function minimizes the prediction error and improves the overall performance of system.

4) Training:
The training phase majorly includes five steps, as follows:
Step 1 (Input pre-processed dataset):
In this step, 90% of the pre-processed dataset is used as input for training the algorithm. The dataset comprises 115 features that have been pre-processed to remove any missing or erroneous values, and standardized using techniques like MinMaxScaler.

Step 2 (Hybrid deep learning phase):
The hybrid deep learning technique is applied in this phase by utilizing two deep learning algorithms, namely DNN and LSTM. These algorithms are executed in parallel on the inputted dataset. Both DNN and LSTM models process the data simultaneously and independently to capture different aspects and patterns within the dataset. This parallel processing can help enhance the model's overall understanding and representation of the data.

Step 3 (Add Layer):
The "Add Layer" step involves adding a layer that combines the outputs of the DNN and LSTM models. This layer takes a list of inputs, which in this case would be the outputs of the DNN and LSTM models, and produces a single tensor as the output. This merging of outputs from different models allows for the integration of diverse information and can potentially improve the overall performance of the system.

Step 4 (Return Sequence):
The "Return Sequence" parameter determines whether the final output in the output sequence or the entire output sequence should be returned. In this case, the default value is set to False, which means only the final output of the merged model is returned. This setting indicates that the system focuses on making a single prediction rather than generating a sequence of predictions.

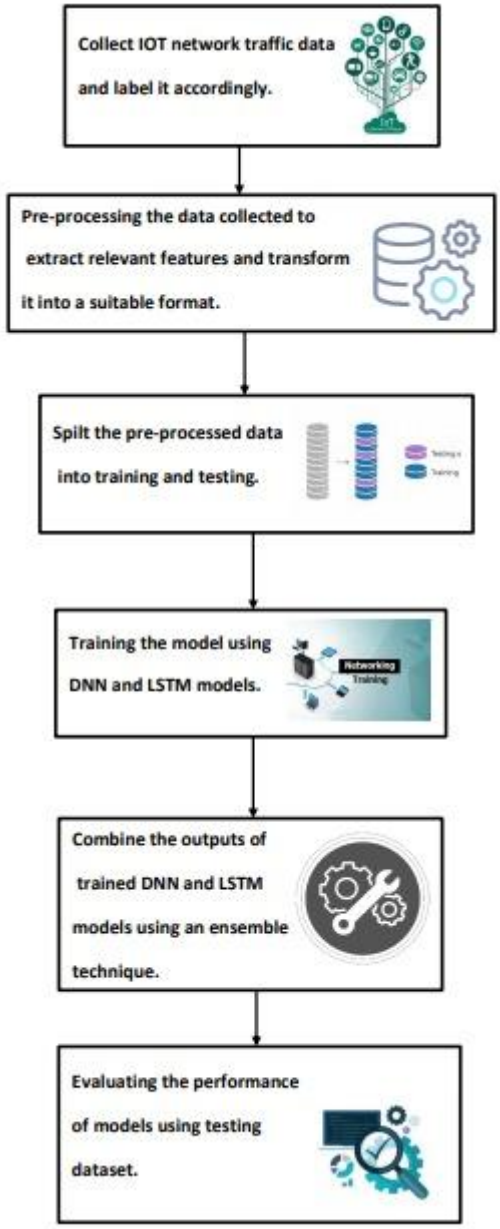
Step 5 (Softmax function):
The softmax function is used as the activation function in the final output layer of the model. Since there are multiple classes to be classified (Benign, GAFGYT, and MIRAI), the softmax function is suitable. It normalizes the output probabilities across all classes, ensuring that the predicted probabilities sum up to 1. By minimizing prediction errors and improving the detection rate, the softmax function aids in accurate classification among the different classes.

5) Testing:
Now, the testing dataset is been given to analyze the working of model. The DL model has been compared to existing algorithms to evaluate the effectiveness in botnet detection. The performance of model is been calculated using metrics such as accuracy, precision, recall, F1 score etc.

The DNN-LSTM algorithm is as follows:
Data training input: There are X network features for every
Y unit of network traffic.
Output: is N for normal or label attacks
For each Yi for N
|
Ci = CNN (Yi) process
|

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

End
For each Ci process
Li = LSTM (Ci) process
|
Merge
|
End
For Each Li Process
N = softmax (Li)
end

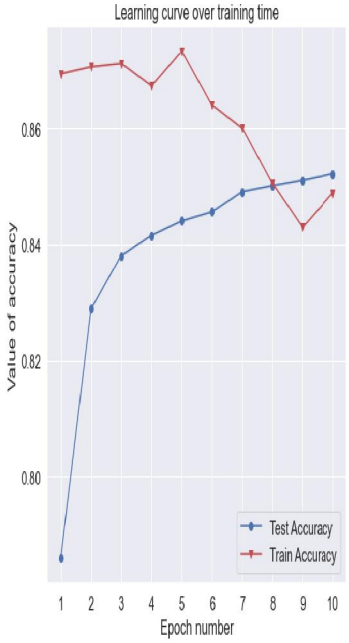


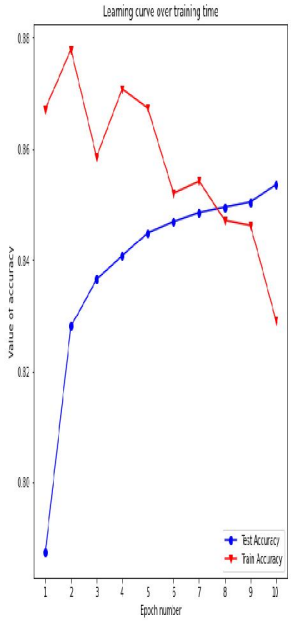
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

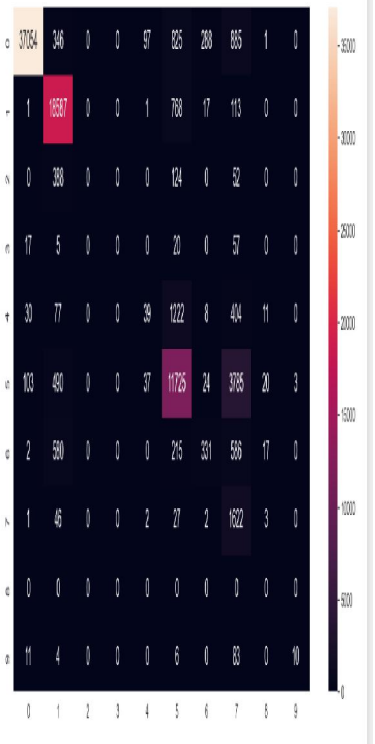
V. RESULTS AND DISCUSSION

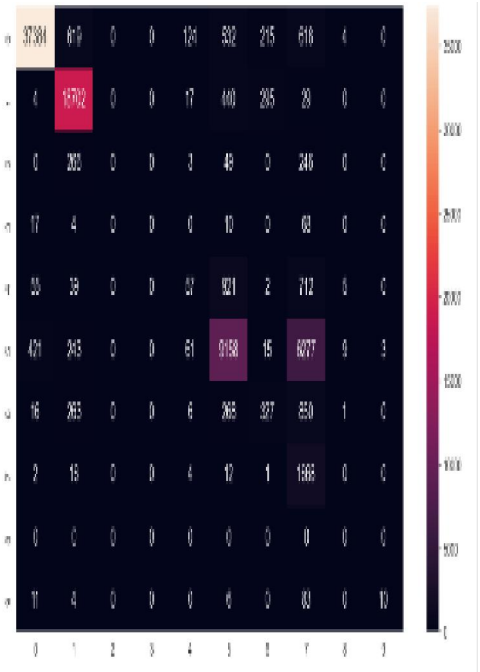
In order to evaluate the effectiveness of our algorithm, we compared it with other algorithms that were developed. We employed a cross-validation technique, specifically the method, which divided the training data into several subsets or "folds." Each time, one subset was used as a test set while the remaining subsets were used for training. This approach helped us gather accurate information about our model's performance by addressing the issues of overfitting and underfitting. Additionally, it provided valuable insights into how our model performs with unforeseen data during the training process.

A 10-fold cross-validation technique is used to obtain average detection accuracy, precision, recall, and F1-score for unbiased results.









IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 11, May 2023

VI. CONCLUSION

The IoT is the latest research topic and the Security issues is also more effective part. In this research, we focused on devices which are not supported by their manufacturers, updates might not be available for those ones. Another challenge to IoT safety is to approve that signal across the web link between expedients and veil services or apps is protected. Encoding the message cannot be done by many IoT devices until they are sent over the network.Next step in research would be inclusion of Secure Shell module to cope with Secure Shell-specific attacks and turning our honeypot into a honeynet that would simultaneously emulate multiple IoT devices.

REFERENCES

[1]. ChamanWijesiriwardana and Prasad Wimalaratne, ‘‘On the Detection and Analysis of Software Security Vulnerabilities’’, 2017.

[2]. R.GaneshBabu, A.Nedumaran, AsefaSisay, ‘‘Machine Learning in IoT Security Performance Analysis of ,2019.

[3]. RanadheerErrabelly, KeweiSha, Wei Wei, T. Andrew Yang, ‘‘EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security’’, 2017.

[4]. Bogdan Oniga Stefan Harsan Farr Adrian MunteanuVasileDadarlat, ‘‘IoT infrastructure secured by TLS level authentication and PKI identity system’’, 2018.

[5]. RuchiVishwakarma, Ankit Kumar Jain."A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks."2019 International Conference on trends in Electronics and Informatics.

[6]. HadeelAlazzam,AbdulsalamAlsmady,Amaal Al Shorman."Supervised Detection of IoT Botnet Attacks."

[7]. Xiaoyu Liang, TaiebZnati."A Long Short-term Memory anabled framework for DDoSDetcetion."

[8]. R. K. Kodali, V. Jain, S. Bose, and L. Boppana, “IoT Based on Smart Security and Home Automation System,” International Conference on Computing, Communication and Automation, pp. 1286-1289, 2016

[9]. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, “Internet of things: a survey and enabling technologies protocols and application” IEEE Communication Surveys & Tutorials, vol. 17, no. 4, Fourth Quarter 2015

[10]. Sriram S, Vinayakumar Ry, MamounAlazabz, Soman KP. "Network Flow based IoT Botnet Attack Detection using Deep Learning."

[11]. G. Apruzzese, M. Colajanni, and M. Marchetti, "Evaluating the effectiveness of adversarial attacks against botnet detectors," in Proc. IEEE Int. Symp. Netw. Comput. Appl., Oct. 2019, pp. 18.

[12]. Z. M. Algelal, E. A. Ghanialdhaher, D. N. Abdul-Wadood et al., "Botnet detection using ensemble classiers of network ow," IAES Int. J. Electr. Comput. Eng., vol. 10, no. 3, p. 2543, 2020.