OPTIMIZATION ACCURACY OF INTRUSION DETECTION FOR CYBERSECURITY SYSTEMS USING AUTOENCODER DEEP LEARNING MODEL

Monika Koli[1], Vinod Kumar Yadav[2], Dr. P. K. Sharma[3]
[1]M.Tech Research Scholar, Computer Science Engineering Department
[2]Associate Professor and HOD of Computer Science Engineering Department
[3]Professor and Principal
[1]NRI Institute of Research and Technology, Bhopal, India
[2]NRI Institute of Research and Technology, Bhopal, India
[3]NRI Institute of Research and Technology, Bhopal, India

Abstract- **Intrusion detection is the process of analyzing packets on a network to determine whether packets are legitimate or illegitimate. The main challenges in this area include the large amount of data used for training and the fast and streamlined data provided for the forecasting process. Also, the conflicting information that exists in the void causes more problems for the access control model. In this article, classification accuracy and parameters of augmented autoencoder deep learning model are compared with traditional deep learning techniques and other machine learning methods. The framework can be used not only to analyze tweets but also to analyze users' perceptions of higher education in India. The proposed framework is based on the deep learning autoencoder model. Since LSTM model can select different values, Augmented Autoencoder deep learning model is used to improve its operation with the help of evolutionary algorithm.**

Keywords- **LSTM, Autoencoder, Intrusion Detection, Deep Learning**

I. Introduction

For a computer to solve a problem, first a suitably effective algorithm is written which can take care of the problem and then, this algorithm is implemented into the hardware or software. The entire problem does not have a direct algorithm and in such cases when the algorithm can't be determined then, this problem can't be solved using a direct programming approach. Machine learning (ML) expands the ability to work with computers by giving a chance to solve problems in cases where algorithms cannot be manually designed. An algorithm can be specified as non-constructive utilizing instances of right behavior [1]. In this way, ML algorithms are defined as a meta-algorithm for making algorithms from information provided that characterizes what they should create. These algorithms give an incredibly better approach for associating with computers by only providing computing data rather than algorithms for computing. Extending the capacity to tackle issues with computers is good enough but not the only reason to study ML. Learning encourages individuals to comprehend what can be practically computed and between them similarly studying computation can educate the understanding of learning. ML as a scientific discipline examines the computational basis of learning [2, 3].

Trying to tackle issues utilizing computational models of learning reveals insight into our comprehension of the mind and at the same time, what we find out about the brain can fill in as motivation for designing models of ML. Studying ML has a scientific value, as an approach to not only to understand computation but at the same time to understand learning. At the same time for science to matter, it should positively affect the world.

Chances to increase the research in the field of ML is by having a positive impact on the world which can be done by continually keeping up an association with imperative practical problems. ML techniques have the capability to solve many numerous particular issues of practical and business interest [4]. As scientists, our only concern is to study science, so maybe we can start with a new method and then find the problems it solves, or we can start with the problem and then take the necessary steps to solve it. In either case, critical research will be undertaken that will enable us to understand the strengths, weaknesses, and limitations of the current system and to highlight the importance of the problem. The main text of this article uses the use of deep learning (DL) to solve cybersecurity problems. The aim of the research project published in this paper is to test the limitations of models based on DL architectures on various problems in cybersecurity, compare the performance with classical ML algorithms, determine how successful the problem is, and maintain the necessary process to do it. They are helpful by introducing new ones that are needed. Overall, the main aim of the research work is to apply DL architectures to a variety of cybersecurity problems that can be considered attractive and realistic [5, 6].

II. Intrusion Detection System

IDS detects malicious computer systems and applies forensics after the attack is complete. Check network resources to identify intrusions and attacks that are not blocked by protection technologies (firewalls, router packet filters, and name servers). Destruction is an effort to think twice about the confidentiality, integrity, or accessibility of a post. Affective recognition processes can be seen as a weak comparison to those who truly understand. Abuse-based IDS (shown in Figure 1) to pre-detect breaches of security regulations. However, everything is complicated by the possibility of unexpected adverse effects [9, 11]. An example would be a designer in an organization that carries a lot of information in a short amount of time. This can be a potential data issue, but the inductive strategy will not be aware of it, as data transfer [12] is allowed. For this particular reason, an imperfect analysis is introduced where the client or framework is analyzed and deviations from the analysis are taken into account. While both frameworks are very useful, transitions between the two can reduce but not eliminate any damage. A key difference in the characteristics of success with IDS is the basis for data analysis. These two principles are network conversations used by network-based IDSs and packets found in organizations used by network-based IDSs. Contracts can be semi-documents, applications or related tools [11].

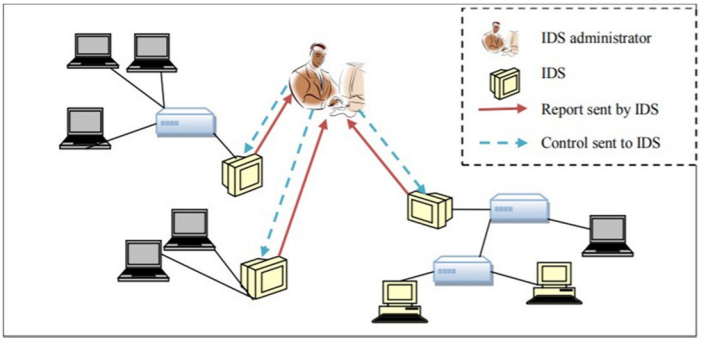


Figure 1: Intrusion Detection System

There are several problems with IDS based on host and network IDS. They include:

- Heterogeneous operating systems make the enumeration of system-specific detection parameters extremely long for any system.
- Increasing the number of basic hubs in the organization expands execution.
- Execution debasement in the host framework because of extra security exercises, for example, registration.
- Difficulty in detecting attacks at the network level.
- Host with insufficient computing power to offer a complete host-based IDS.

Interestingly, network-based interruption recognition frameworks can have a focal framework with an organization association with latently screen network traffic. They no affect framework execution and can without much of a stretch recognize network-level assaults when introduced at the edge of the organization. Network-based ID implementation is too simple [13]. Host based IDs in a critical performance-sensitive host network must be carefully selected so as not to unduly restrict the performance of each system.

III. Proposed Methodology

Normally, neural networks operate as a "black box," making decisions based on inputs. Information on learning experiences is stored using weights in static memory. The LSTM network was introduced to offer explicit representation for memory in RNNs. These models are an adaption of RNNs and are best suited for sequential input. In the network, the memory unit is referred to as a "cell." on this research proposal, we propose to test the efficacy of LSTM for sentiment classification of brief texts with distributed representation on social media. Figure 2 illustrates how the algorithm functions.

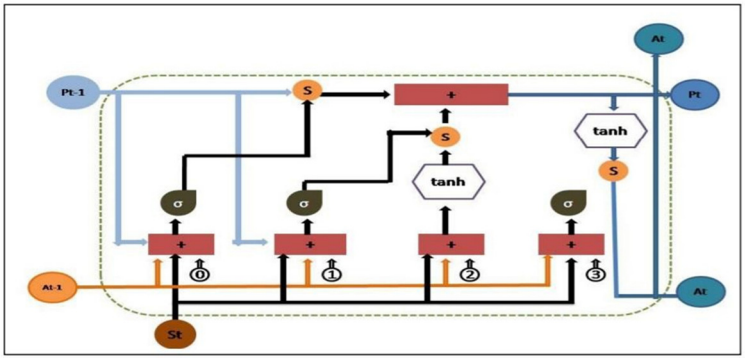


Figure 2: Working of LSTM

The LSTM network accepts three inputs at, "St," "At-1," and "Pt-1," as shown in Fig. 2 above. The input vector for the current time step is called "St." The output or hidden state passed from the prior LSTM unit is designated as "At-1."

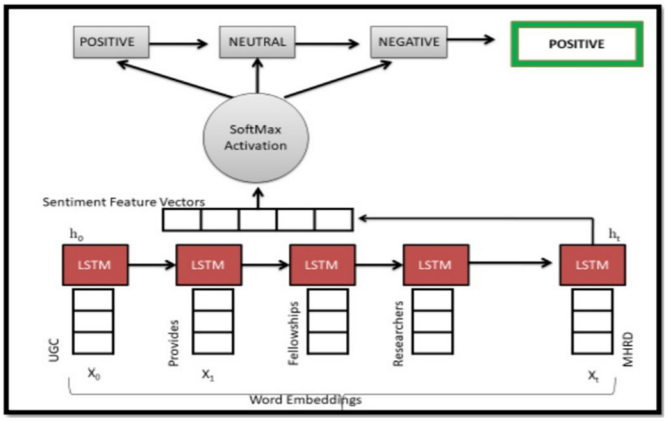


Figure 3: Sentiment Classification using LSTM

And 'Pt-1' is the memory element or cell state of the previous unit. It has two outputs such as, 'At' and 'Pt', where, 'At' is the output of the current unit and 'Pt' is the memory element of the current unit. Every decision is made after considering current input, previous output and previous memory information. When the current

output is obtained the memory is updated. The 'S' indicates the 'Forget' element of multiplication. When the value for the forget element is given as '0' it forgets ninety percent of old memory. For all other values such as 1, 2, and 3 a fraction of old memory is allowed by the unit. The plus operator is present for the piece wise summation to summarize old and new memory. The amount of old memory is decided by the 'S' sign. As a result of two operations, Pt-1 is changed to Pt. The activation functions described in the fig. 3 are the sigmoid and tanh activation functions having output as a forget valves. The second activation valve is termed as new memory element as it includes old memory while processing new inputs. The old memory, previous output and current input along with a bias vector decides the amount of memory to be given as input to the next unit.

Algorithm: Sentiment analysis on Twitter data using Enhanced LSTM Input: Twitter data set with class labels
Output: Classification of tweets whether tweet implies positive, negative or neutral sentiment.
Step 1: Pre-handled tweets taken as .csv document, data set is loaded
Step 2: Tagging the tweets
Step 3: Tagged tweets converted into vectors (word2vector conversion)
Step 4: Apply Evolutionary algorithm on the vectors to select the best feature set
Step 5: Enhanced-LSTM performs training only on the best features set selected by Evolutionary Algorithm and obtains a Model
Step 6: Testing data set is supplied to the Model obtained by Enhanced LSTM
Step 7: Evaluate the performance of this model based on some parameters

AutoEncoder Deep Learning Model:-
The input and output of feedforward neural networks that use autoencoders are identical. They reduce the input's dimension before using this representation to recreate the output. The code, also known as the latent-space representation, is an efficient "summary" or "compression" of the input. Encoder, code, and decoder are the three parts of an autoencoder. The input is compressed by the encoder, which also creates a code. The decoder then reconstructs the input exclusively using the code.

Autoencoders primarily function as dimensionality reduction (or compression) algorithms and have the following key characteristics:

- Data-specific: Autoencoders are only able to meaningfully compress data similar to what they have been trained on. Since they learn features specific for the given training data, they are different than a standard data compression algorithm like gzip. So we can’t expect an autoencoder trained on handwritten digits to compress landscape photos.
- Lossy: The output of the autoencoder will not be exactly the same as the input, it will be a close but degraded representation. If you want lossless compression they are not the way to go.
- Unsupervised: To train an autoencoder we don’t need to do anything fancy, just throw the raw input data at it. Autoencoders are considered an unsupervised learning technique since they don’t need explicit labels to train on. But to be more precise they are self-supervised because they generate their own labels from the training data.

IV. Simulation Result

Step 1: Collect the dataset, this dataset contains intrusion website information.
Step 2: Performing EDA on the dataset and get to know that it can be done as binary classification and multi-class classification.

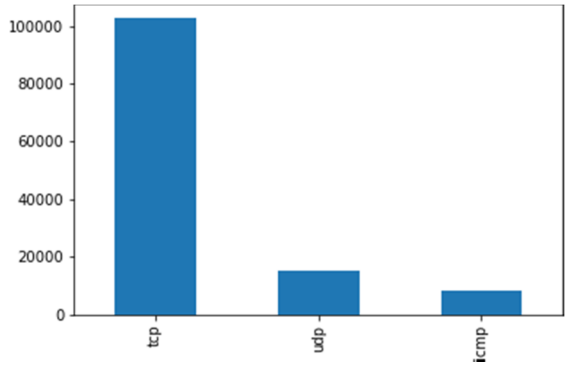
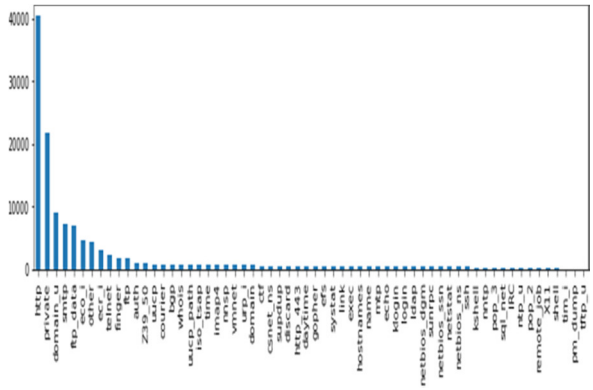
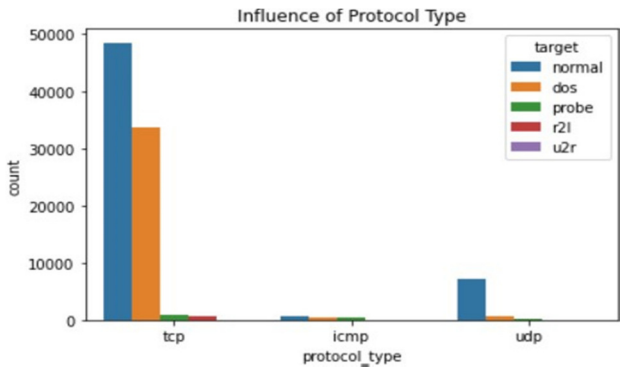
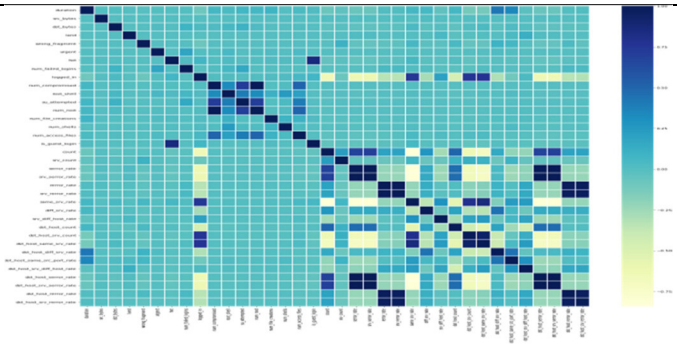
Step 3: Processing

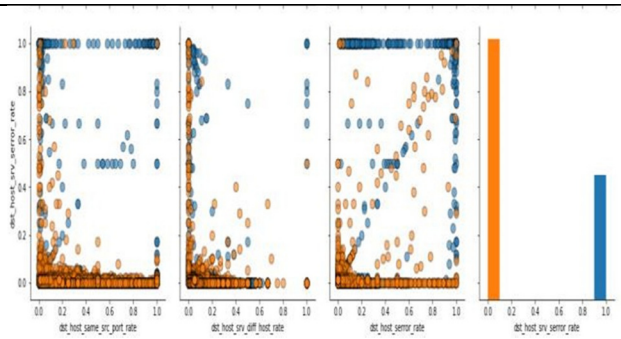
- Dropping Null values.
- Removing duplicate Values
- Changing To scalar values
- Feature Extraction

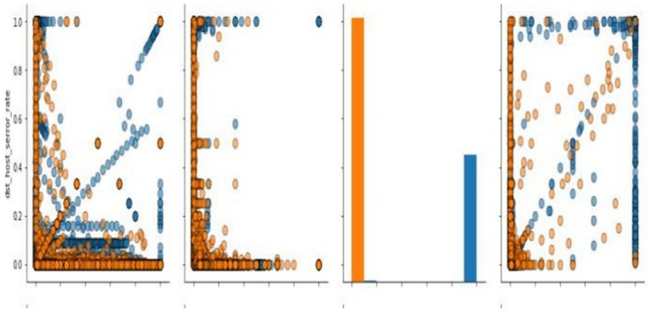
Step 4: Plotting graphs and done final processing on the data for the training.
Step 5: Creating an AutoEncoder Deep Learning model and fitting the data to it, let it train. After completion, use the model for testing.
Step 6: Evaluation of the model, testing the model on the test set and measuring the performance in terms of precision, recall and F1-Score. The AutoEncoder Deep learning model performed very well.

Pre-Processing:

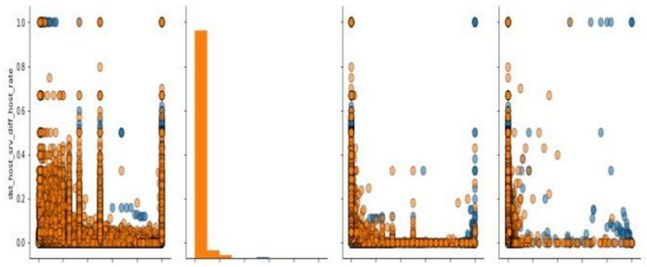
Table 1: Pre-Processing Parameter

1. Bar Graph protocol type 	**2.** Every Service Graph 
3. Protocol type influence on target 	**4.** Correlation between whole data 
5. Dst_host_port	**6.** dst_host_serror_rate

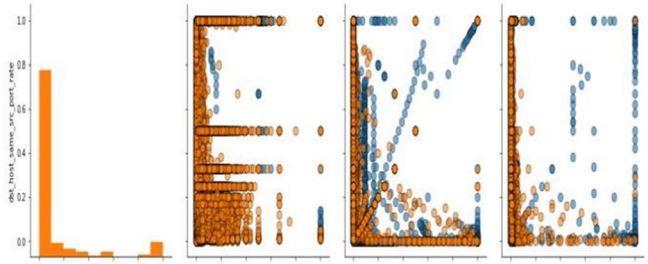




7. dst_host_srv_diff_host_rate



8. dst_host_same_src_port_rate



9. Pre-processing:

- Dropping Null values.
- Removing duplicate Values
- Changing To scalar values
- Feature Extraction

10. Model Summary

a. Optimization Function = adam

b. Loss function = mean_squared_err

Encoding dim = 50

```
Model: "model_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_19 (InputLayer)        [(None, 93)]              0
_________________________________________________________________
dense_51 (Dense)             (None, 50)                4700
_________________________________________________________________
dense_52 (Dense)             (None, 93)                4743
=================================================================
Total params: 9,443
Trainable params: 9,443
Non-trainable params: 0
_________________________________________________________________
```

Table 2: Comparison Parameter

Model	Accuracy	Precision	Recall	F1-Score
Base Alex	**0.82**	**0.83**	**0.82**	**0.81**
Base LSTM	**0.78**	**0.78**	**0.78**	**0.75**
Base XGBoost	**0.77**	**0.81**	**0.77**	**0.73**
Proposed	**0.88**	**0.86**	**0.88**	**0.90**

Accuracy, loss & Result Graph:

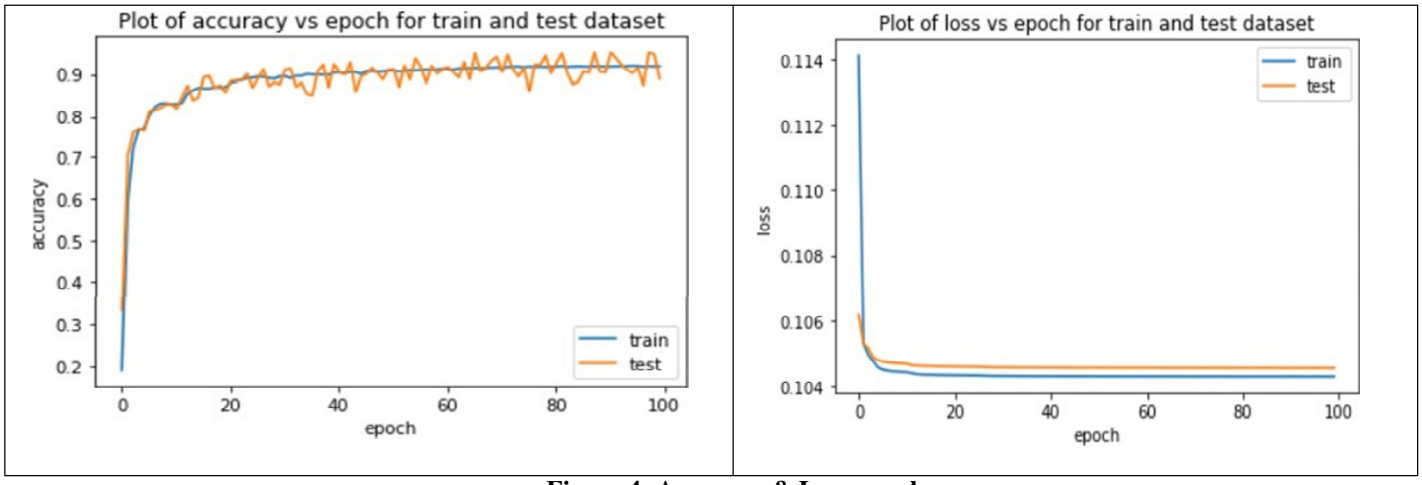


Figure 4: Accuracy & Loss graph

ROC Curve:

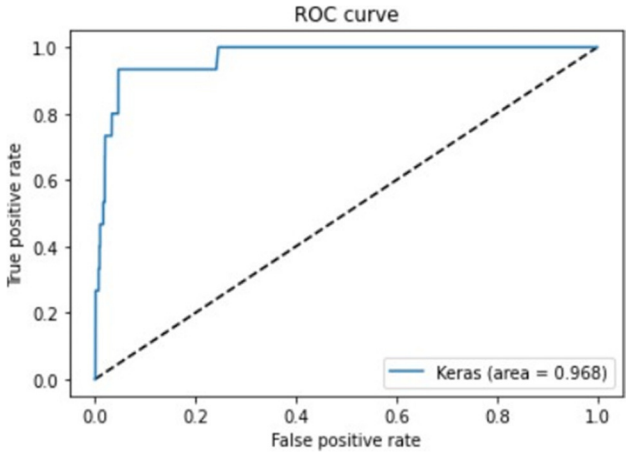


Figure 5: ROC Curve

V. Conclusion

The accuracy of grouping engineered and constant data using ML and DL calculations is examined and evaluated. Ordinary DL algorithms, such as Random Forest, SVM, and XG-Boost, performed well against created data but failed to address real-time data. An innovative deep learning computation called Enhanced LSTM was put forth to organise massive amounts of continuous data and provided the greatest results on real-time information when compared to deep learning models.

References

[1] Lan Liu, Pengcheng Wang, Jun Lin, and Langzhou Liu, “Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning”, IEEE Access 2020.

[2] A. Raghavan, F. D. Troia, and M. Stamp, ``Hidden Markov models with random restarts versus boosting for malware detection,'' *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 2, pp. 97107, Jun. 2019.

[3] Zhiyou Zhang and Peishang Pan "A hybrid intrusion detection method based on improved fuzzy C-Means and SVM", IEEE International Conference on Communication Information System and Computer Engineer (CISCE), pp. no. 210-214, Haikou, China 2019.
[4] Afreen Bhumgara and Anand Pitale, "Detection of Network Intrusion Using Hybrid Intelligent System", IEEE International Conferences on Advances in Information Technology, pp. no. 167-172, Chikmagalur, India 2019.
[5] Azar Abid Salih and Maiwan Bahjat Abdulrazaq "Combining Best Features selection Using Three Classifiers in Intrusion Detection System", IEEE International Conference on Advanced science and Engineering (ICOASE), pp. no. 453-459, Zakho - Duhok, Iraq 2019.
[6] Lukman Hakim and Rahilla Fatma Novriandi "Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset", IEEE International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), pp. no. 330-336, Jember, Indonesia 2019.
[7] T. Sree Kala and A. Christy, "An Intrusion Detection System Using Opposition Based Particle Swayam Optimization Algorithm and PNN", IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, pp. no. 564-569, Coimbatore, India 2019.
[8] L. A. Maglaras, K.-H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras, and T. J. Cruz, "Cyber security of critical infrastructures," ICT Express, vol. 4, no. 1, pp. 42–45, 2018.
[9] A. Ahmim, M. Derdour, and M. A. Ferrag, "An intrusion detection system based on combining probability predictions of a tree of classifiers," International Journal of Communication Systems, vol. 31, no. 9, p. e3547, 2018.
[10] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," arXiv preprint arXiv: 1812.09059, 2018.
[11] Z. Dewa and L. A. Maglaras, "Data mining and intrusion detection systems," International Journal of Advanced Computer Science and Applications, vol. 7, no. 1, pp. 62–71, 2016.
[12] B. Stewart, L. Rosa, L. A. Maglaras, T. J. Cruz, M. A. Ferrag, P. Simões, and H. Janicke, "A novel intrusion detection mechanism for scada systems which automatically adapts to network topology changes." EAI Endorsed Trans. Indust. Netw. & Intellig. Syst., vol. 4, no. 10, p. e4, 2017.
[13] I.Sharafaldin, A.H.Lashkari, andA.A.Ghorbani,"Toward generating a new intrusion detection dataset and intrusion traffic characterization." in ICISSP, 2018, pp. 108–116.
[14] M. A. Ferrag, L. Maglaras, H. Janicke, and R. Smith, "Deep learning techniques for cyber security intrusion detection : A detailed analysis," in 6th International Symposium for ICS & SCADA Cyber Security Research (ICS-CSR 2019), Athens, 10-12 September, 2019.
[15] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35 365– 35 381, 2018.
[16] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of networkbased intrusion detection data sets," *Computers & Security*, 2019.
[17] G.Loukas, E.Karapistoli, E.Panaousis, P.Sarigiannidis, A.Bezemskij, and T. Vuong, "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Networks*, vol. 84, pp. 124– 147, 2019.
[18] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning- based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.
[19] 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*, 2019.
[20] D.S.Berman, A.L.Buczak, J.S.Chavis, and C.L.Corbett,"A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, 2019.