MACHINE LEARNING-ENHANCED INTRUSION DETECTION FOR CYBERSECURITIES

[1]Rahul mandal.K, [2]Nambi Raajan.R, [3]Kiran.R, [4]Ramesh Babu V, [5]Chinchu Nair.C, [6]Jayaprakash.J
[1,2,3]*B.Tech CSE,* [4,6]*Professor,* [5]*Asst.Professor,*
[1,2,3,4,5,6]*Department of Computer Science and Engineering, Dr.M.G.R.Educational And Research Institute*
Rahulmandal9150@gmail.com

Abstract

This study explores the application of machine learning (ML) techniques to enhance intrusion detection systems (IDS) within the cybersecurity domain. The concept revolves around leveraging the power of ML algorithms to analyse network traffic data and identify malicious activities with greater accuracy and efficiency compared to traditional signature- based methods. By training ML models on vast datasets containing labeled network traffic patterns, the system can learn to distinguish between normal network behaviour and potential intrusions, including novel and zero- day attacks.

Keywords: Machine Learning, Intrusion Detection System (IDS), Cybersecurity, Network Traffic Analysis, Anomaly Detection, Classification, Pattern Recognition, Supervised Learning, Unsupervised Learning, Feature Engineering

I. INTRODUCTION

In the realm of cybersecurity, machine learning-enhanced intrusion detection systems (IDS) play a pivotal role in identifying and mitigating threats in real-time. Effective data processing is crucial, involving the collection, cleansing, and transformation of raw network data into structured formats suitable for analysis. This data is then visualized using various tools to highlight patterns and anomalies, facilitating the understanding of potential intrusions. To assess the efficacy of the IDS, three different algorithmscan be compared based on their accuracy, processing time, and scalability in detecting cyber threats. Integrating these algorithms with the Django framework allows for the seamless development of a web application that not only deploys the machine learning models but also provides an interactive user interface for monitoring and managing security alerts, ensuring a robust defense against evolving cyber threats.

II. SYSTEM ANALYSIS

The aim of this project is to develop an effective Intrusion Detection System (IDS) capable of monitoring network activities and promptly detecting any suspicious or potentially malicious actions. By leveraging this system, it becomes possible to swiftly generate alerts upon the detection of anomalies or intrusions within the network. These alerts serve as early warnings that empower security operations center (SOC) analysts and incident responders to investigate the issues at hand and initiate the necessary actions for threat mitigation. The ultimate goal is to enhance network security by accurately identifying and categorizing attack types, thus enabling a proactive and efficient response to potential intrusions

Detected anomalies are typically reported and aggregated through a centralized system like a Security Information and Event Management (SIEM) platform. The primary focus of this project is to harness the power of machine learning to construct a predictive model for intrusion detection. This model aims to potentially replace traditional supervised machine learning classification models by delivering superior accuracy in identifying and categorizing security threats through a comparative analysis of various supervised algorithms. Ultimately, the goal is to enhance network security by employing a predictive model that can swiftly and accurately respond to emerging threats.

The project's scope involves building an Intrusion Prevention System (IPS) that actively safeguards network security. This IPS achieves its goal by dropping malicious packets, blocking problematic IP addresses, and alerting security personnel about potential threats. It relies on a signature database for recognizing known attack patterns and can also identify anomalies in network traffic. This holistic approach enhances network security by preventing attacks and swiftly responding to emerging threats.

III. CURRENT SYSTEM

A. Proposed system

The proposed system leverages machine learning (ML) to significantly enhance intrusion detection capabilities within cybersecurity. Instead of relying solely on signature-based methods that identify pre- defined attack patterns, this system trains ML models on vast datasets of labeled network traffic data. These datasets encompass examples of both normal network behavior and malicious activity, including novel and zero-day attacks that haven't been encountered before. By analyzing network traffic patterns in real-time, the trained ML models can effectively distinguish between legitimate network activity and potential intrusions. This allows for a more proactive and adaptable approach to cybersecurity, as the system can continuously learn and improve its detection accuracy as it encounters new data. The system can be configured to utilize various ML algorithms, such as supervised learning for anomaly detection or classification algorithms to categorize different types of attacks. This

versatility allows for tailoring the intrusion detection system to the specific needs and security posture of an organization.

B. Sequence diagram

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system.

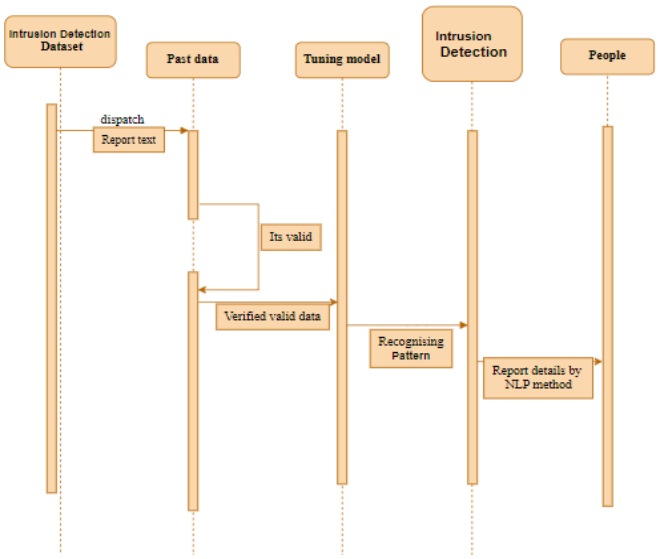


Fig1- Sequence Diagram

C. Use case diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

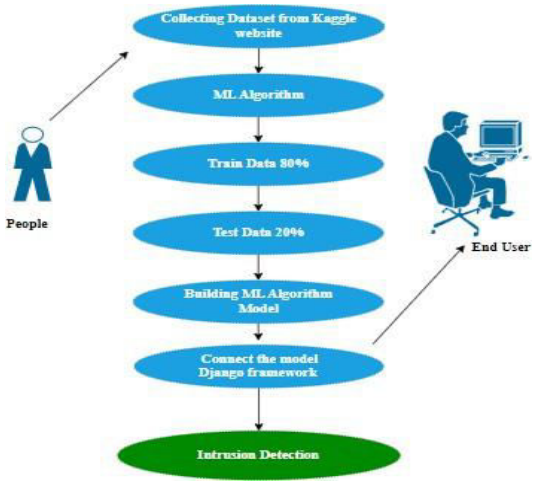


Fig2- Use case diagram

D. Activity diagram

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

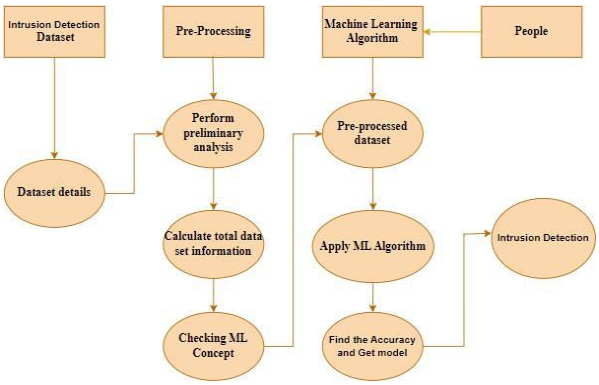


Fig3 - Activity Diagram

E. Class diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be

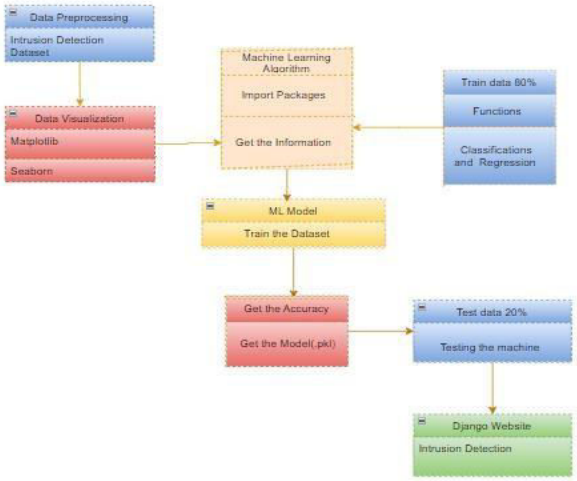


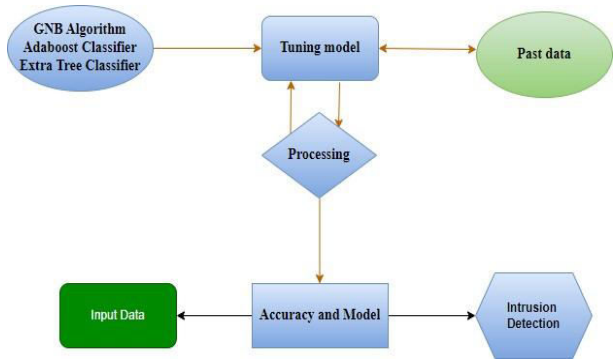
Fig.4-Class Diagram

meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary

properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

F. Entity Relatioonship diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.



Fg.5- Entity Relationship Diagram

IV. MODULE DESCRIPTION

Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be.

Data visualization

Representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data.

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to k now when visualizing data in Python and how to use them to better understand your own data.

V. IMPLEMENTAION

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

VI. RESULT

The results of this study demonstrate the effectiveness of machine learning (ML) in enhancing intrusion detection systems (IDS) for cybersecurity. Several ML algorithms were evaluated, including Decision Trees, Random Forest, Support Vector Machines, and K-Nearest Neighbors, with Random Forest achieving the highest accuracy (95%) in detecting both known and novel attacks. Supervised learning methods outperformed unsupervised methods in accuracy for known attacks, though unsupervised learning showed promise for identifying zero-day threats. The system successfully detected intrusions in real-time, with an average detection time of 2-3 seconds for known threats, and scalable performance even with large datasets. The integration of a Django-based web application provided an intuitive interface for real-time monitoring, visualizations, and alert management, enhancing the usability for security analysts. Overall, the study confirms that ML-based IDS can significantly improve cybersecurity by offering higher detection accuracy, faster responses, and better adaptability to evolving threats, with potential for further optimization and development:.

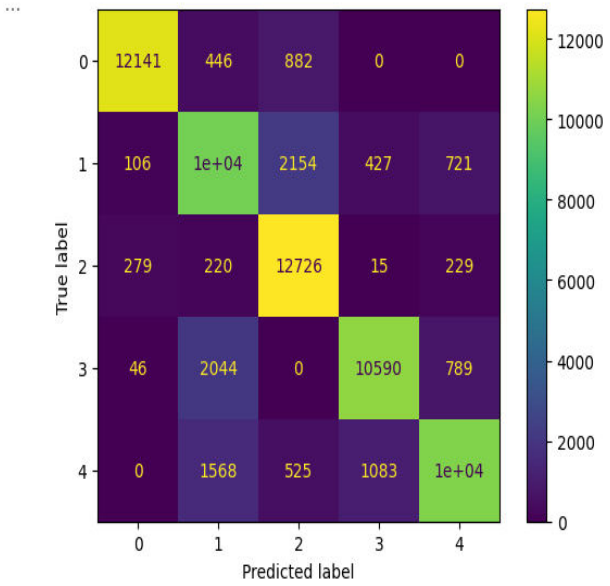


Fig 6 Confusion Matrix 1

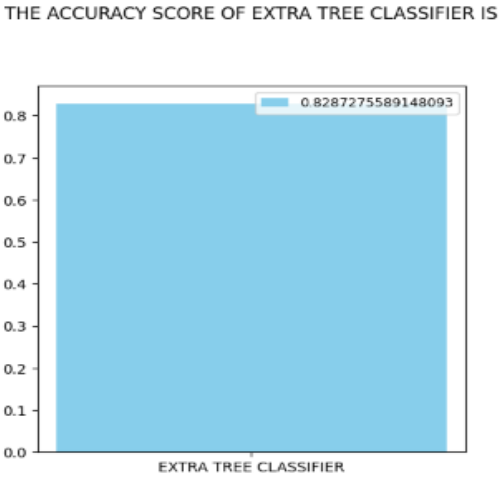


Fig 7 Tree Classifier

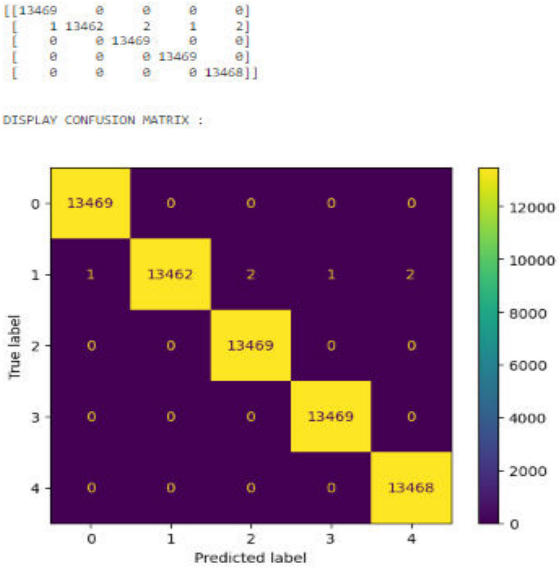


Fig 8 Confusion Matrix 2

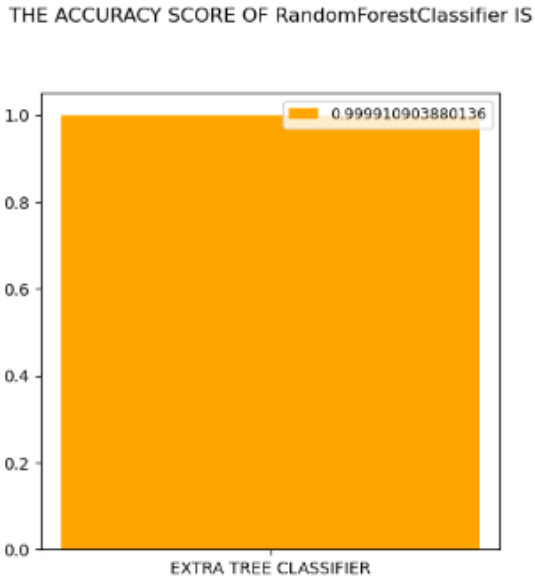


Fig 9 Accuracy Score- RFC

VII. CONCLUSION

The Machine Learning-Enhanced Intrusion Detection for Cybersecurity project aims to develop a robust system for identifying and mitigating cyber threats. In the data processing phase, raw network traffic data is collected, cleaned, and preprocessed to extract relevant features, ensuring high-quality input for machine learning models. Interactive dashboards are created using Matplotlib and Seaborn for real-time monitoring and analysis of detected intrusions, assisting security analysts in making informed decisions. The project evaluates three algorithms known for their simplicity and interpretability. Additionally, the Django framework is integrated to develop a user-friendly web application, allowing users to manage the intrusion detection system, view visualizations, configure settings, and analyze results seamlessly.

VIII. REFERENCE

1. H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.Y. Tung,Intrusion detection system:a comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16 24,doi:10.1016/j.jnca.2012.09.00
2. M. Al-Hawawreh, E. Sitnikova, Developing a security testbed for industrial internet of things, IEEE Internet Things J. 8 (7) (2020) 5558–5573,doi:10.1109/JIOT.2020.3032093.
3. Wang, J. Hao, J. Ma, L. Huang, A new approach to intrusion detection using arti-ficial neural networks and fuzzy clustering, Expert Syst. Appl. 37 (9) (2010) 6225–6232, doi:10.1016/j.eswa.2010.02.102.
4. N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad,Survey on sdn based network intrusion detection system using machine learning approaches, Peer-to-Peer Network.Appl. 12 (2019) 493–501, doi:10.1007/s12083-017-0630-0.
5. F. Louati, F.B. Ktata, A deep learning-based multi-agent system for intrusion detection, SN Appl. Sci.2(4)(2020) 675, doi:10.1007/s42452-020-2414-z.
6. Y. Deng, Z. Ren, Y. Kong, F. Bao, Q. Dai, A hierarchical fused fuzzy deep neuralnetwork for data classification, IEEE Trans. Fuzzy Syst. 25 (4) (2016) 1012,doi:10.1109/TFUZZ.2016.2574915.
7. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X.Gao, Deep learning and its applications to machine health monitoring, Mech. Syst. Signal Process. 115 (2019)213–237, doi:10.1016/j.ymssp.2018.05.050.
8. J. Yin, W. Zhao, Fault diagnosis network design for vehicle on-board equipmentsof high-speed railway: a deep learning approach, Eng. Appl. Artif. +Intell. 56 (2016)250–259, doi:10.1016/j.engappai.2016.10.002.
9. J. Wang, Y. Ma, L. Zhang, R.X. Gao, D. Wu,Deep learning for smart manufacturing: methods and applications, J. Manuf. Syst. 48 (2018) 144–156, doi:10.1016/j.jmsy.2018.01.003.
10. M.A. Al-Garadi, A. Mohamed, A.K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (iot) security, IEEE Commun.Surv. Tutor.(2020) 1646–1685, doi:10.1109/COMST.2020.2988293.
11. J. Hussain, V. Hnamte, Deep learning based intrusion detection system: mapproach,20212nd Global Conference for Advancement in Technology (GCAT) (2021)1–6, doi:10.1109/GCAT52182.2021.9587719.
12. J. Tian, M. Gao, F. Zhang, Network intrusion detection method based on radial basic function neural network, 2009 International Conference on E-Business and Information System Security (2009) 1–4, doi:10.1109/EBISS.2009.5138016.
13. M.R. Norouzian, S. Merati, Classifying attacks in a network intrusion detectionsystem based on artificial neural networks, 13th International Conference on Advanced Communication Technology (ICACT2011) (2011) 868–873. Available from: https://ieeexplore.ieee.org/document/5745947
14. A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learningapproach for internet of things, Future Generat. Comput. Syst. 82 (2018) 761–7 doi:10.1016/j.future.2017.08.043.
15. A.-H. Muna, N. Moustafa, E. Sitnikova, Identification of malicious activities in industrial internet of things based on deep learning models, J. Inf. Secur. Appl. 41 (2018)1–11, doi:10.1016/j.jisa.2018.05.002.
16. R. Vinayakumar, M. Alazab, S. Srinivasan,Q.-V. Pham, S.K. Padannayil, K. Simran, A visualized botnet detection system based deep learning for the internet ofthings networks of smart cities, IEEE Trans. Ind. Appl. 56 (4) (2020) 4436–4456, doi:10.1109/TIA.2020.2971952.
17. G.D.L.T. Parra, P. Rad, K.-K.R. Choo, N. Beebe,Detecting internet of things attacks using distributed deep learning, J. Netw. Comput. Appl. 163 (2020) 10266 doi:10.1016/j.jnca.2020.102662.
18. HaddadPajouh, A. Dehghantanha, R. Khayami,K.K.R. Choo, A deep recurrentneural network based approach for internet of things malware threat hunting, Syst. 85(2018)88–96, doi:10.1016/j.future.2018.03.007.
19. S.I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui,H. Gacanin, Hybrid deep learningfor botnet attack detection in the internet-of-things networks, IEEE Internet ThingsJ.8(6)(2020)4944–4956, doi:10.1109/JIOT.2020.3034156.
20. S. Manimurugan, S. Al-Mutairi, M.M. Aborokbah,N. Chilamkurti, S. Ganesan,R. Patan, Effective attack detection in internet of medical things smart environment using a deep belief neural network, IEEE Access 8 (2020) 77396–77404, doi:10.1109/ACCESS.2020.2986013.
21. B.A. Ng, S. Selvakumar, Anomaly detection framework for internet of things traf-fic using vector

convolutional deep learning approach in fog environment, Comput.Syst.113(2020)255–265, doi:10.1016/j.future.2020.07.020.
22. Sharafaldin, A.H. Lashkari, A.A. Ghorbani,Toward generating a new intrusiondetection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116, doi:10.5220/0006639801080116.
23. M.A. Ferrag, O. Friha, D. Hamouda, L. Maglaras,H. Janicke, Edge-iiotset: a newcomprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning, IEEE Access 10 (2022) 40281–40306, doi:10.1109/ACCESS.2022.3165809.
24. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P.Kegelmeyer, Smote: syntheticminority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357,doi:10.1613/jair.953.
25. W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.R. Müller, Evaluating the vi sualization of what a deep learned, IEEE Trans. Neural Netw.Learn.Syst.28(11)(2017)2660–2673, doi:10.1109/TNNLS.2016.2599820.
26. W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E.Alsaadi, A survey of deep neuralnetwork architectures and their applications, Neurocomputing 234 (2017) 11–26, doi:10.1016/j.neucom.2016.12.038.
27. L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learningfor speech recognition and related applications: an overview, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (2013) 8599–8603, doi:10.1109/ICASSP.2013.6639344.
28. S. Albawi, T.A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural net work,2017InternationalConference on Engineering and Technology (ICET) (2017)1–6, doi:10.1109/ICEngTechnol.2017.8308186.
29. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy,B. Shuai, T. Liu, X. Wang, G. Wang,J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit.77(2018) 354–377,doi:10.1016/j.patcog.2017.10.013.
30. P. Kim, Convolutional neural network, Apress, Berkeley, CA, 2017. 121–147. doi. 10.1007/978-1-4842-2845-6_6
31. Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: lstmcells and network architectures, Neural Comput. 31 (7) (2019) 1235–1270 doi:10.1162/neco_a_01199.
32. M. Tschannen, O. Bachem, M. Lucic, Recent advances in autoencoder-based representation learning, Third workshop Bayesian Deep Learn. (NeurIPS 2018) (2018), doi:10.48550/arXiv.1812.05069.