Design And Implementation Of Network Traffic Classifier

[1]Namita Parati , [2]Dr Salim Y. Amdani

[1,2]Department of CSE, Babasaheb Naik College of Engineering, Pusad, Maharashtra, India.

Abstract. Despite the large number of research efforts that applied specific machine learning algorithms for network traffic classification, recent work has highlighted limitations and particularities of individual algorithms that make them more suitable to specific types of traffic and scenarios. The new improvement in industry computerization and associated gadgets made a tremendous interest for network assets. Conventional organizations are turning out to be less successful to deal with this enormous number of traffic created by these advancements. Simultaneously, Software characterized organizing (SDN) presented a programmable and versatile systems administration arrangement that empowers Machine Learning (ML) applications to computerize networks. Issues with conventional strategies to order network traffic and dispense assets can be tackled by this SDN arrangement. Network information assembled by the SDN regulator will permit information investigation strategies to examine and apply AI models to alter the organization the executives.

Keywords: Network, Traffic, Security, Tool, classification.

I. Introduction:

One of the main challenges in automating the detection and classification of anomalies in modern computer networks is the fact that different anomalies present diverse spatio-temporal network traffic characteristics; as such, a single detection and classification process is unlikely to be effective [1]. It is additionally asset escalated and really infeasible to exactly portray all peculiarities of an area, since the arrangement of irregularities doesn't continue as before; new inconsistencies ordinarily arise when framework spaces develop with new highlights, upgrades, and fixes. In any case, on the grounds that new elements keep on showing up over the long run, peculiarity location frameworks ought to be adequately adaptable to oblige new circumstances, rather than being confined to a consistent arrangement of predefined abnormalities. One of the methodologies that have been utilized to adapt to this situation is the utilization of AI based classifiers [2]. Research in the space shows that peculiarities can be identified, somewhat, by base classifiers separately [1]. For traffic classifiers to obtain new abilities and adjust to various conditions, they ought to gain from past encounters as opposed to considering each disengaged order task. This figuring out how to-learn (meta-learning) [3] approach is a basic advance for accomplishing adaptable traffic classifiers. This subject has become particularly alluring in view of the reason that meta-classifiers are regularly more exact than the singular classifiers that make them up [4]. The area of meta-learning is otherwise called gathering learning [5]. Group learning incorporates a wide scope of examination endeavors that look to track down the best techniques to construct mixes of classifiers [6]. Works, for example, [7] and [8] began to examine meta-learning procedures with regards to organize traffic arrangement. Notwithstanding, they considered a more restricted set of meta-learning strategies. In this paper, we present a relative report between various meta-learning procedures and individual classifiers inside the extent of organization traffic. Accordingly, we can decide the best strategy to be utilized in this specific circumstance. The classifiers are utilized to recognize ordinary from assault traffic from an informational index containing genuine traffic information. For this, we chose four meta-learning strategies ordinarily introduced in the writing, characterized the base classifiers to be utilized, analyzed the exhibition of these methods among one another, and furthermore the execution of similar base classifiers when utilized exclusively.

II. Related Work

In the field of artificial intelligence, related researchers apply machine learning algorithms to the classification of network traffic, for example, applying limited Boltzmann machines to the grouping of DoS traffic [12], utilizing Artificial Neural Network (ANN) to recognize the vindictive traffic [13], the utilization of profound conviction network in network traffic characterization [14], etc. Since the organization traffic information itself likewise has expected fleeting and spatial highlights, the worldly component is reflected in the current and past rush hour gridlock streams, and the spatial element is reflected in the topological relationship between's the traffic streams. Thusly, the spatial and worldly highlights additionally impact the acknowledgment of ordinary and strange traffic. Applicable scientists have applied CNN to the spatial component extraction of organization traffic, and have accomplished specific accomplishments [15], [16]. Riyaz and Ganapathy [11] proposed an element choice technique in light of restrictive arbitrary fields and direct relationship coefficients to choose the most contributing highlights, and afterward utilized the CNN model for additional component extraction to work on the exhibition of organization traffic acknowledgment. Xu et al. [15] proposed the LSTMs-AE model, which joins LSTM with the Auto-Encoder (AE). The model uses LSTM's time series include extraction capacity and AE's element portrayal learning capacity to further develop execution. Azizjon et al. [16] utilized the 1D-CNN model for managed learning of organization traffic transient highlights, and through examinations to check that its exhibition is superior to customary AI models like arbitrary timberland and SVM. In the wake of preprocessing the first traffic information, Xu [17] utilized picture handling innovation to change over traffic information into grayscale pictures, and afterward utilized CNN to convolve the grayscale pictures of traffic to separate the relationship between's highlights. Ling [18] handled the spatial highlights of the information by utilizing numerous CNNs with various scale convolution parts, and joined with LSTM to remove transient elements. Imrana et al. [19] proposed the bidirectional LSTM (BidL STM) model for the grouping of strange traffic, and checked its presentation to be preferable over LSTM and different models. Applying LSTM to the extraction of organization traffic highlights can actually separate the time series highlights between traffic streams. Albeit the use of CNN to the extraction of traffic spatial elements additionally has a specific presentation improvement, but CNN is more appropriate for handling Euclidean primary information like pictures. The type of organization traffic information is generally a one-layered structure, and the spatial connection between traffic streams is more like a geography structure. Diagram convolution model [20] has a decent element extraction capacity for topological design and has been generally applied in certain fields. Zhao et al. [21] proposed a blend of chart convolutional network and Gated Recurrent Unit (GRU) to extricate the transient and spatial highlights of traffic streets and make more precise expectations of street traffic stream. The outcomes show that its exhibition is superior to conventional time series relapse models like ARIMA and SVR. Yao et al. [22] build a solitary text chart for the corpus in view of word co-event and record word relationship, and afterward become familiar with the text diagram convolutional network for the corpus. Contrasted and different strategies, the presentation of this model is more conspicuous. By examining the application status and restrictions of the above works, the chart convolution model is as yet in the exploratory stage. In the field of organization security, the use of diagram convolution model in network traffic include extraction has significant examination importance.

Network traffic classification has been widely covered in the writing. In fixed networks, a few stream based strategies have been proposed to characterize traffic continuously by utilizing the principal bundles of the stream (early characterization) [19], [20] or disconnected in light of the entire stream (late order). These methodologies have additionally been stretched out to remote organizations, by utilizing the capacity of SL to distinguish application fingerprints. In [21], a gadget fingerprinting plan in view of learning traffic examples of foundation exercises is proposed. The strategy utilizes support vector and k-closest neighbors classifiers, prepared with information from 20 clients with various mixes of applications associated with a 3G organization. In [22], six kinds of versatile applications are distinguished by examining the parcel size and transmission course of the initial 20 parcels as info highlights of a secret Markov model. In [23], a structure for fingerprinting and recognizable proof of portable applications is introduced in view of choice trees and backing vector classifiers prepared with measurable stream highlights assembled in view of timing and objective IP address/port. In [23], a similar system is utilized to evaluate the corruption of grouping execution because of changes in application fingerprints. In [24], a gathering approach joining unique cutting edge classifiers is proposed. Four classes of blend methods are thought about, varying in acknowledged classifiers' results, preparing prerequisites and learning plan. Approval on a dataset of genuine client action shows higher exactness contrasted with the singular utilization of the considered classifiers.

III. Network Classifier:

a. **Dataset Input and Pre-processing** :-The real source of KDD (Knowledge Discovery in Databases) dataset is engaged for collecting the dataset. For the scrutinization of the KDD dataset, the train set contains 78% records in train set and 75% in test set. The studying algorithm is partial on the way to the records as enormous count of inefficacious records in the train set. The techniques have increased detection rates on the continuous records assist in attaining biased outcomes considering the records in the test set. Moreover, this work does the execution 21 study machines for which the complexity level of records are analyzed in KDD for assigning labels to the records of entire training and testing sets of KDD (Knowledge Discovery in Databases). Hence, 21 declaration labels are supplied for every record. The KDD dataset has unstable form for preprocessing the dataa so that the data is cleaned. Meanwhile, under sampling is utilized for cleaning the input dataset. This technique is helpful to remove the inefficacious factors, to clean the set.

b. **Data Collection**: Classically, historical data has been a very important knowledge base for constricting machine learning models [4]. A plentiful and comprehensive set of conceptions about an issue has potential to upgrade the performance and generality of these paradigms. However, this factor is very important in the field of traffic classification due to several reasons. Some of these reasons include the complexity and scalability of web networks, the continual growth of traffic, and privacy rules not allowing the data collection. The phase of data collection allows the measurement of various conditions over the network. This phase mostly gathers IP runs within a timeframe. Moreover, this block consists of many tasks including packet management, flow reconstruction, and storage. It is essential to collect the historical dataset in offline flow. The online run, in contrast, constantly treats the packets' flow.

c. **Feature extraction:** Appropriate features are extracted following the recording of the data that represents the problem. It is a vital step as it permits to measure or compute features that might contain information concerning the process status. Briefly, a feature extraction scheme calculates various metrics reflecting exclusive features in the collected data. Obtaining descriptors that better illustrate the issue is the major objective. The feature extraction process provides output as a structured table generated by feature columns. Every row is a pattern, with an extra random column representing each sample's current position (usually called a label or class). The patterns are not labelled when the status is not known.

d. **Feature selection and reduction**: This step makes use of either feature selection or feature reduction schemes to treat resultant attributes to obtain less space or a set of new features. This is a voluntary process that allows to select or reduce the number of features extracted.

Feature reduction is for creating new features using the original features, whereas feature selection is for finding a reduced set of attributes that better defines a procedure. These steps are intended to reduce issues, e.g., time expenditure and the obscenity of size and so on. These methods are usually classified into Filters, Wrappers and Embedded Schemes, which in turn can be devised by machine learning algorithms.

e. **Classification**: A novel dataset is generated from the original dataset on the basis of selected attributes. The offline run makes the utilization of the new dataset for developing build models using which classification and regression tasks can be performed among other things. The Algorithm Selection block includes procedures and techniques for selecting the most adequate ML (machine learning) model. This approach is extensively executed for discovering various solutions with the implementation of several ML models. For a variety of ML methods, it is essential to discover the best model for classifying the traffic.

IV. Classifier Evaluation Parameters:

i. **Recall**: Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances.

$$\text{Recall (R)} = \frac{TP}{TP + FN}$$

ii. **Precision**: Precision is the fraction of relevant instances among the retrieved instances.

$$\text{Precision (P)} = \frac{TP}{TP + FN + ND}$$

iii. **F1 score**: F1 Score (F-score or F-measure) is a measure of a test's accuracy

$$\text{F-Measure (F1)} = \frac{2 * P * R}{P + R}$$

Where,

TP = True Positive = Number of correctly detected results
FN = False Negative = Number of incorrectly detected results
ND = Not Detected = Number of not detected results
(TP + FN) = Total Detected results
(TP + FN +ND) = Actual Number of available results

V. Results:

Table 1. Network Data Analysis

Sr. No.	Feature/Parameters for Comparison	Network-1	Network-2	Network-3	Network-4

1	Large No. of Iterative Computation	Yes	Yes	Yes	Yes
2	Scalability	Horizontal	Horizontal	Horizontal	Horizontal
3	Mode of Computation	Disk-Based	In-Memory	Disk-Based	In-Memory
4	Mode of Processing	Batch & Stream	Batch & Stream	Batch	Stream
5	Auto-Scaling	Yes	No	Yes	No
6.	Processing Time for Big-Data Set	Fastest	Slower	Less Faster	Not specified
7.	Processing time for Small Data-Set	Faster	Less Faster	Slower	Not Specific
8.	Processing Time for Cluster Size	Fast	Slow	Slow	Not compared
9.	Processing time for sending a tweet messages of 100Kb per message.	Slow	Fast	Less faster	Not compared
10.	Processing time for sending a tweet messages of 1000Kb per message.	Fast	Slow	Less slower	Not specific
11.	CPU Consumption(Batch Mode)	High CPU Usage	Low CPU Usage	Medium CPU Usage	Not compared
12.	CPU Consumption (Stream Mode)	Medium CPU Usage	Not compared	Low CPU Usage	High CPU Usage
13.	CPU Consumption (Batch Mode)	High CPU Usage	Low CPU Usage	Medium CPU Usage	Not compared
14.	CPU Consumption (Stream Mode)	Not compared	High CPU Usage	Low CPU Usage	Low CPU Usage
15.	Latency (RAM3S Framework)	High latency	Moderate	Low	Very Low
16.	Throughput	Moderate Throughput	High Throughput	Less Throughput	Very Less Throughput
17.	Sustainable Input rate (Local & Cloud Cluster)	Not Compared	Low Input rate	Very high Input rate	Moderate Input rate
18.	Execution Time Required for big-data set	High Execution Time	High Execution Time	Low Execution Time	Less Execution Time

20.	Ranking of Page	Worse	Good	Better	Not Compared
21	Sorting, grepping, and Connected Component	Best	Best	Worst	Good
22.	Scalability (Big Data-Graph Processing)	Not Compared	Best	Good	Worse
23.	Scalability (Large Data-Set and Fixed no. of node)	Excellent	Better	Worse	Good
24.	Fault Tolerance	Less	High	Very High	Good
25.	Execution Time for Tera Sorting	Very fast	Fast	Slow	Not Compared

Table 2. Traffic Classification Evaluation

Network	**Severity**	**Recall**	**Precision**	**Accuracy**
Network-1	Mild	79.8	79.5	82.1
	Moderate	86.5	81.5	88.6
	Severe	80.2	83.5	80.3
Network-2	Mild	79.9	80.6	81.7
	Moderate	83.9	72.1	89.8
	Severe	86.8	80.6	79.9
Network-3	Mild	82.4	81.6	83.9
	Moderate	83.5	80.4	88.4
	Severe	80.6	81.9	80.7
Network-4	Mild	72.1	82.6	78.6
	Moderate	80.6	78.9	86.5
	Severe	81.6	79.6	80.5

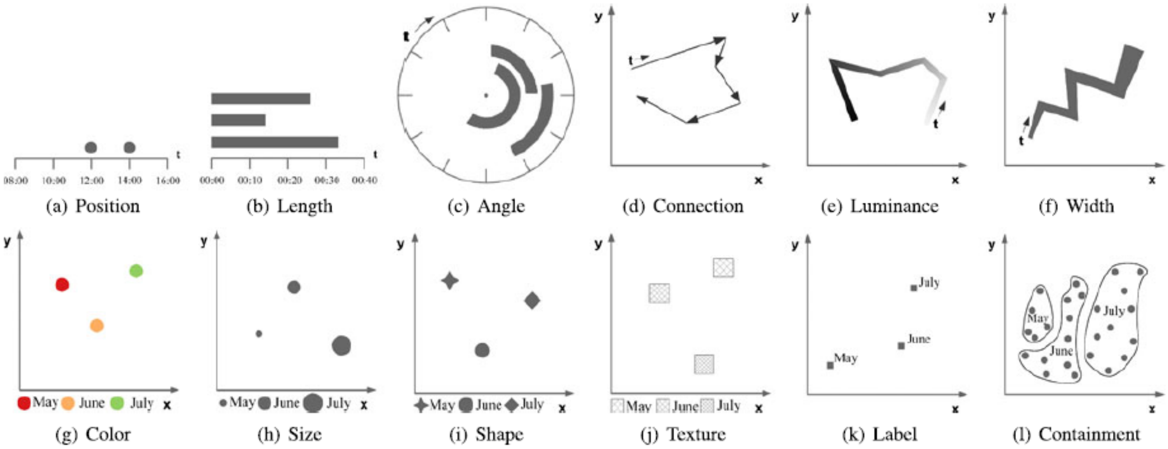
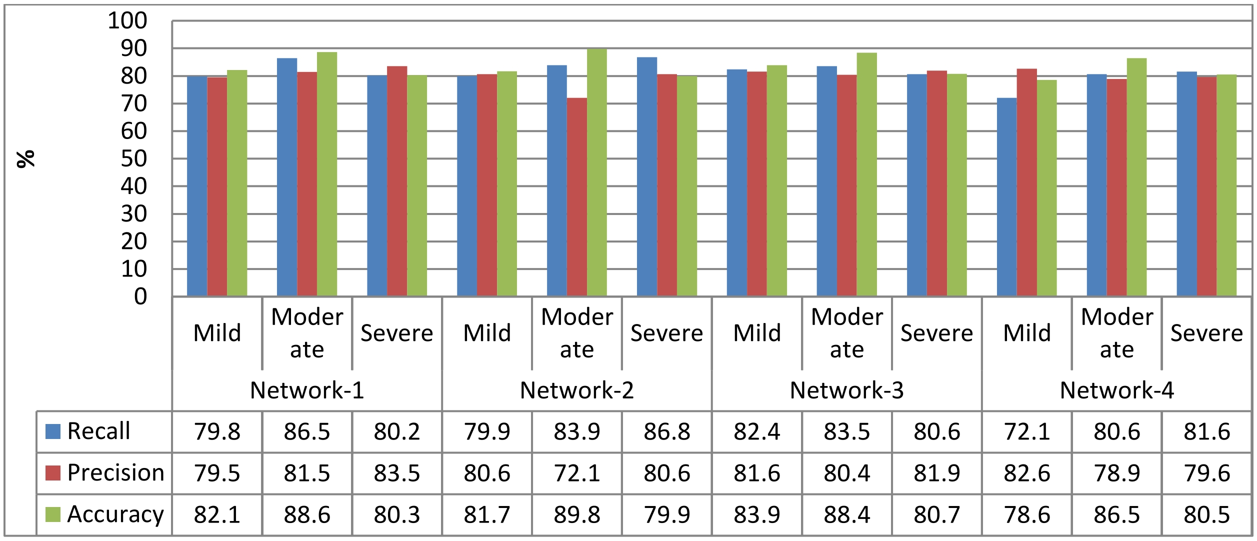


Figure 1. Visualization



	Network-1			Network-2			Network-3			Network-4		
	Mild	Moderate	Severe	Mild	Moderate	Severe	Mild	Moderate	Severe	Mild	Moderate	Severe
🟦 Recall	79.8	86.5	80.2	79.9	83.9	86.8	82.4	83.5	80.6	72.1	80.6	81.6
🟥 Precision	79.5	81.5	83.5	80.6	72.1	80.6	81.6	80.4	81.9	82.6	78.9	79.6
🟩 Accuracy	82.1	88.6	80.3	81.7	89.8	79.9	83.9	88.4	80.7	78.6	86.5	80.5

Figure 2. Network based Classification and Evaluation

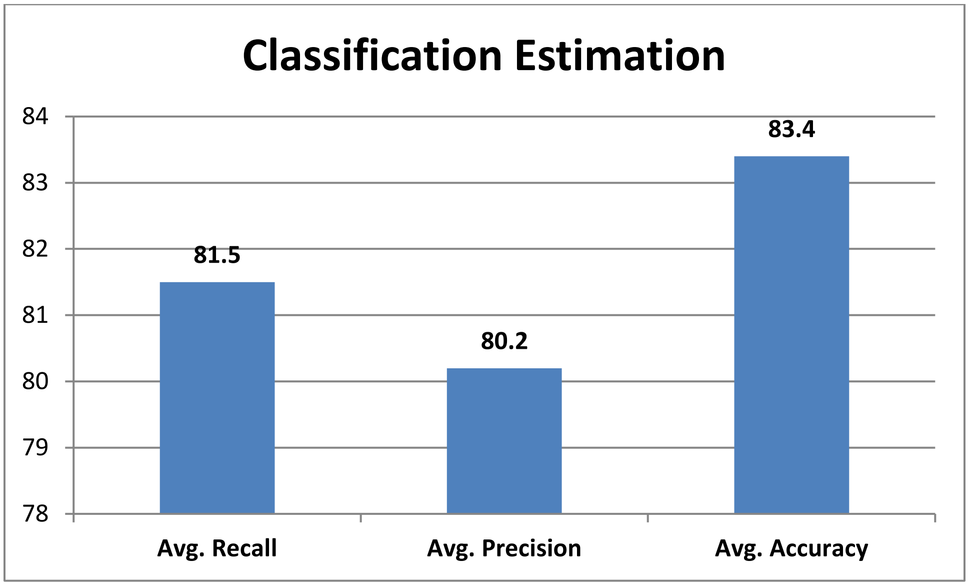


Figure 3. Classification Estimation

VI. Conclusion:

The network traffic classification techniques posses the three phases namely port-based, payload-based and flow statistics-based. The process to classify the network traffic is deal with recognizing distinct types of applications or traffic data-part for which the obtained data packets scrutinized that is crucial in the transmission of networks of real global world. The traditional port-based course of action based on contributing the standard ports that the famed functions deploy. The traffic can be categorized in several stages phases instance as pre-processing, to draw out the attributes and to succeed in doing the classification. This research work makes the usage of voting classification algorithm in way to classify the traffic. The suggested algorithm supply the greater precision, accuracy and recall in contrast to the existing SVM (Support Vector Machine) algorithm.

References:

[1] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1153–1176, Second quarter 2016.

[2] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba, F. E. Solano, and O. M. C. Rendon, “Machine learning for cognitive network management,” IEEE Communications Magazine, vol. 56, no. 1, pp. 158–165, 2018.

[3] P. K. Chan and S. J. Stolfo, “Experiments on multi strategy learning by meta-learning,” in Proceedings of the Second International Conference on Information and Knowledge Management, ser. CIKM ’93. New York, NY, USA: ACM, 1993, pp. 314–323.

[4] S. Dzeroski and B. ˇ Zenko, “Is combining classifiers with stacking better ˇ than selecting the best one?” Machine Learning, vol. 54, no. 3, pp. 255– 273, Mar 2004.

[5] C. Zhang and Y. Ma, Ensemble Machine Learning: Methods and Applications. Springer Publishing Company, Incorporated, 2012.

[6] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” Journal of Internet Services and Applications, vol. 9, no. 1, p. 16, Jun 2018.

[7] J. M. Reddy and C. Hota, “P2p traffic classification using ensemble learning,” in Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop, ser. I-CARE ’13. New York, NY, USA: ACM, 2013, pp. 14:1–14:4.

[8] C. Wang, X. Guan, and T. Qin, “A traffic classification approach based on characteristics of sub flows and ensemble learning,” in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), May 2017, pp. 588–591.

[9] A. S. da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, “Atlantic: A framework for anomaly traffic detection, classification, and mitigation in sdn,” in NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, April 2016, pp. 27–35.

[10] L. Didaci, G. Giacinto, and F. Roli, “Ensemble learning for intrusion detection in computer networks,” in AI*IA, Workshop on ”Apprendimento automatico: metodi e applicazioni”, Siena, Italy, 11/09/2002 2002.

[11] S. Seo, S. Park, and J. Kim, ‘‘Improvement of network intrusion detection accuracy by using restricted Boltzmann machine,’’ in Proc. 8th Int. Conf. Comput. Intell. Commun. Netw. (CICN), Tehri, India, Dec. 2016, pp. 413–417.

[12] A. Shenfield, D. Day, and A. Ayesh, ‘‘Intelligent intrusion detection systems using artificial neural networks,’’ ICT Exp., vol. 4, no. 2, pp. 95–99, Jun. 2018.

[13] I. Sohn, ‘‘Deep belief network based intrusion detection techniques: A survey,’’ Expert Syst. Appl., vol. 167, Apr. 2021, Art. no. 114170.

[14] Y. Yang, Research on Convolutional Neural Network Intrusion Detection Model Based on Network Traffic Feature Map. Hangzhou China: Hangzhou Dianzi Univ., 2020.

[15] S. Z. Lin, Y. Shi, and Z. Xue, ‘‘Character-level intrusion detection based on convolutional neural networks,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[16] Y. Xu, Y. Tang, and Q. Yang, ‘‘Deep learning for IoT intrusion detection based on LSTMs-AE,’’ in Proc. 2nd Int. Conf. Artif. Intell. Adv. Manuf., Oct. 2020, pp. 64–68.

[17] M. Azizjon, A. Jumabek, and W. Kim, ‘‘1D CNN based network intrusion detection with normalization on imbalanced data,’’ in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC), Feb. 2020, pp. 218–224.

[18] Y. Xu, ‘‘A research of intrusion detection based on image processing within the framework of deep learning,’’ M.S. thesis, Univ. Electron. Sci. Technol. China, Chengdu, China, 2020.

[19] Y. Ling, Research on Intrusion Detection System Model Based on Deep Neural Network. Hangzhou, China: Hangzhou Dianzi Univ., 2020.

[20] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, ‘‘A bidirectional LSTM deep learning approach for intrusion detection,’’ Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524.

[21] T. N. Kipf and M. Welling, ‘‘Semi-supervised classification with graph convolutional networks,’’ 2016, arXiv:1609.02907.

[22] L. Zhao, Y. Song, C. Zhang, Y. Liu, and H. Li, ‘‘T-GCN: A temporal graph convolutional network for traffic prediction,’’ IEEE Trans. Intell. Transp. Syst., vol. 21, no. 9, pp. 3848–3858, Sep. 2019.

[23] L. Yao, C. Mao, and Y. Luo, ‘‘Graph convolutional networks for text classification,’’ in Proc. AAAI Conf. Artif. Intell., 2019, vol. 33, no. 1, pp. 7370–7377.

[24] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, ‘‘Traffic classification on the fly,’’ SIGCOMM Comput. Commun. Rev., vol. 36, no. 2, pp. 23–26, 2006.

[25] Y. Liu, J. Chen, P. Chang, and X. Yun, ‘‘A novel algorithm for encrypted traffic classification based on sliding window of flow’s first n packets,’’ in Proc. 2nd IEEE Int. Conf. Comput. Intell. Appl. (ICCIA), Sep. 2017, pp. 463–470.

[26] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, ‘‘Who do you sync you are?: Smartphone fingerprinting via application behaviour,’’ in Proc. 6th ACM Conf. Secur. Privacy Wireless Mobile Netw. (WiSec), 2013, pp. 7–12.

[27] I.-C. Hsieh, L.-P. Tung, and B.-S.-P. Lin, ‘‘On the classification of mobile broadband applications,’’ in Proc. IEEE 21st Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD), Oct. 2016, pp. 128–134.

[28] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, ‘‘Robust smartphone app identification via encrypted network traffic analysis,’’ IEEE Trans. Inf. Forensics Security, vol. 13, no. 1, pp. 63–78, Jan. 2018.

[29] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, ‘‘Multi-classification approaches for classifying mobile app traffic,’’ J. Netw. Comput. Appl., vol. 103, pp. 131–145, Feb. 2018