2021 International Symposium on Networks, Computers and Communications (ISNCC) | 978-1-6654-0304-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISNCC52172.2021.9615806

An Entropy-based Hybrid Mechanism for Large-Scale Wireless Network Traffic Prediction

Guilherme N. N. Barbosa[1], Martin Andreoni Lopez[2], Dianne S. V. Medeiros[1], Diogo M. F. Mattos[1]
[1]LabGen/MídiaCom – PPGEET/TET/TCE – Universidade Federal Fluminense – UFF – Niterói, Brazil
[2]Technology Innovation Institute (TII) – Abu Dhabi, United Arab Emirates
{guilhermenasseh,diannescherly,diogo_mattos}@id.uff.br, martin@ssrc.tii.ae

Abstract—**The rising of the Internet of Things (IoT) applications fosters the exponential increase of smart devices, expanding the Internet's attacking surface. Anomaly prediction mechanisms are mandatory to anticipate security threats. Besides, traffic monitoring and prediction models deliver more resilient and efficient network services. This paper proposes a lightweight user-behavior prediction mechanism based on the decomposition of the network traffic features' entropy through Discrete Wavelet Transform (DWT) applied to network-flow Shannon Entropy's time series. The DWT decomposes the entropy into linear and nonlinear components. We compare two forecasting models using Long Short Term Memory (LSTM) Networks and Auto-Regressive Integrated Moving Averages (ARIMA). We evaluate our mechanism in a large-scale academic wireless network, with more than 500 access points. LSTM performs up to 10 times better than ARIMA for predicting the real value of nonlinear flow-source entropy. Considering the transport protocol entropy, LSTM is up to 8 times better than ARIMA, and our results show a high entropy value. LSTM also outperforms ARIMA concerning the prediction time, which is 42% lower for LSTM's worst-case training time than ARIMA's best-case training time.**

Keywords—**Network Traffic, Prediction, ARIMA, Shannon Entropy, Network Security, LSTM, Discrete Wavelet Transform**

I. Introduction

The related technology to the Internet of Things (IoT) continuously evolves and has achieved a maturity level that contributes to reduce the production-cost of mobile and smart devices. Several initiatives also encourage the adoption of IoT solutions, which increases mobile data generation. Recent studies show that in 2017, 23% of the global Internet traffic was from smartphones [1]. The mobile traffic over the Internet traffic ratio is estimated to reach 50% in 2022, with tablets and machine-to-machine (M2M) communication corresponding to 14% of the data. Shortly, approximately 66% of global network traffic will rely on mobile communication technologies.

The IEEE 802.11 standards, the WiFi technology, are one of the key enabling technologies for the IoT [2]. These communication technologies are widely deployed and commonly cover large areas, such as university *campi* [3]. In such a scenario, it is essential to study mobile data's growth due to its impact on network management and security, especially when its size and density increase. Thus, traffic analysis is a crucial tool to set management and designing goals for the network infrastructure. In this paper, traffic analysis relies on user-related features identified in the network flows to predict usage patterns and network behavior [4].

We propose a network-entropy behavior prediction mechanism for large-scale wireless networks.The entropy is a powerful measurement of randomness degree of an information system. Entropy allows detecting usage-pattern changes. The main contributions of this paper are three-fold: i) extracting information from categorical features, such as IP addresses and accessed services using the Shannon Entropy; ii) proposing a noise-resistant approach to predict behavior, as we decompose each analyzed time series into linear and non-linear components; and iii) applying the Discrete Wavelet Transform (DWT) on time series for traffic prediction. Moreover, we evaluate two distinct approaches to predict linear and non-linear components for each feature, a Neural Network based on the Long Short-Term Memory (LSTM) and the Auto-Regressive Integrated Moving Averages (ARIMA).Both methods use few computational resources, and do not require many parameters to configure. Furthermore, we compare the performances of an stochastic model and a neural network model.

Previous works focus on comparing wavelet decomposition and random forest approaches [5], or on testing the use of neural networks to predict traffic behavior [6]. Conversely, our proposal differs on forecasting the entropy as an approach to predict the degree of dispersion in the network-flows information. The variation of such dispersion can indicate a behavior anomaly, as network-flow characteristics are mostly random, especially features such as destination IP and source port. Our proposal's evaluation runs over a dataset containing real traffic data from the institutional wireless network of the *Universidade Federal Fluminense*, which is the largest federal university in number of undergraduate students in Brazil. The network daily accounts for more than 5,000 users simultaneously connected at the peak hours [7]. Results show that entropy is a reliable metric to predict network-related categorical features. We highlight that signal decomposition improves the prediction models since it reduces the training dataset, delivering minimal prediction errors.

The remainder of the paper is organized as follows. Section II presents the related works. In Section III, we define the problem of network traffic prediction. Section IV presents the proposed mechanism. Our results are presented in Section V. Finally, Section VI concludes this work, and presents future research directions.

II. Related Work

Different prediction techniques have been proposed to identify and anticipate suspicious traffic, anomalous behaviors, and network misuse patterns. Other techniques focus on calculating

This work is partially funded by CNPq, CAPES, FAPERJ, FAPESP (2018/23062-5), City hall of Niterói/FEC/UFF (PDPA 2020) and RNP.
978-1-6654-0304-7/21/$31.00 ©2021 IEEE

the traffic entropy to improve the quality of service that Internet Service Providers (ISPs) offer. Giotis *et al.* compare different approaches using native OpenFlow and SFlow, and combine both to detect anomalies in Software-Defined Networking (SDN) environments, also using the traffic entropy measurements. They further propose a mechanism to mitigate anomaly events [8]. Bartos *et al.* state prioritizing alerts and correlating alerts are challenging tasks [9]. They claim that the probability that a recently discovered attack occurs again in a short period is high. The authors propose a machine-learning algorithm to estimate the probability that an entity, *i.e.*, a host or a network, becomes a source of the attack soon. To this end, the authors create the Future Misbehavior Probability (FMP) score. The score's goal is to introduce some knowledge, from different sources, about a specific entity, network, or host. The score represents one of these entities' expected behavior, based on machine learning, and then assign a value to it, which predicts future events. In this scenario, the authors analyze two models: Neural Networks and Gradient Boosted Decision Tree (GBDT). They initially compare the Brier Score values, which is a factor in measuring probabilistic forecasts' accuracy. Both neural networks and the GBDT performed well, achieving Brier Score values close to zero. The GBDT, however, performed better.

Wang *et al.* develop a framework to detect and mitigate Link-Flooding Attacks (LFAs) in SDN [10]. Such attacks are hard to detect because the malicious traffic is very similar to legitimate traffic. The attack aims to saturate the communication links to degrade the possible paths that lead to a given service. The proposed framework is composed of three modules: to detect the LFA, to assess whether the attack is an LFA, and to mitigate the impact of the attack, performing traffic engineering to balance traffic throughout the nodes.

Javed *et al.* present an approach to identify botnet attacks based on network traffic and temporal features for connected vehicles. The proposal's primary goal is to select optimal features from input data and analyze the contribution of temporal features on the attack. The authors also compare Adaboost method and other machine learning techniques [11].Holgado *et al.* present an approach to predict traffic attacks using IDS alerts. The goal is to predict each step of an attack using Hidden Markov Model (HMM) detecting similar phases before specific attack occurs. Viterbi and forward-backward algorithms are used to detect whether the attack is underway. [12].

Yang proposes an algorithm to detect network traffic anomalies based on entropy and applying a Support Vector Machine (SVM) classifier [13]. The proposal focuses on detecting anomalies in cloud-computing network traffic using entropy measurements and machine learning. The proposed algorithm calculates and normalizes the entropy, as well as selects the best parameters for the SVM classifier. The author uses the Quantum-behaved Particle Swarm Optimization (QPSO) to optimize parameter selection.

Unlike previous work, we focus on analyzing the decomposed network traffic into linear and non-linear components. Our analysis compares the forecast error for each component using both Long Short-Term Memory (LSTM) neural network and Auto-Regressive Integrated Moving Averages (ARIMA). Our results show that the entropy of the considered network traffic primarily behaves as a non-linear signal, and the LSTM approach is the one that predicts with the lowest error the network behavior and demands lower training time.

III. Traffic Prediction in Large-Scale Wireless Networks

The network traffic prediction problem relates to modeling traffic volume between nodes in a network. Network traffic prediction aims to anticipate network flow characterization that will happen in future time [14]. Besides forecasting the traffic volume, the network traffic prediction problem also deals with protocol classification and protocol distribution forecasting. The protocol classification problem consists of a series of protocol types, determining which protocols are estimated to appear in the network at future steps. An extension of the problem involves forecasting the distribution of network packet features. However, the prediction of network traffic data depends mainly on the data's statistical nature and chronological dependence. Self-similarity and the highly non-linear nature of the network data are statistical characteristics that particularly harden forecasting traffic. Poisson or Gaussian distributions insufficiently model the non-linear nature of the data. Moreover, from the data dependency perspective, network traffic is characterized by long-term autocorrelation, which most statistical models fail to capture [14].

Previous work uses mathematical models of the Auto-Regressive Integrated Moving Average (ARIMA) to predict traffic growth. The model typically runs for offline traffic analysis, and, therefore, the complexity of the predictor is not critical. However, in an online forecasting scenario, separating traffic into its components, such as trends, bursts, and noise, should follow a statistical approach for feasibility and, then, each component separately predicts the traffic. Neural networks are also solutions for online prediction of network traffic, as the artificial neural network finds complex patterns in incoming data. The wavelet transform is suitable for multiscale prediction, as it naturally transforms a signal in multiple resolutions. Applying wavelet transformation enables to reveal the detailed local trends [15].

IV. Hybrid Network Traffic Prediction Mechanism

The proposed mechanism conjugates traffic decomposition using wavelet transform and predictions based on ARIMA and LTSM neural networks. Our mechanism ingests NetFlow monitoring data. Thus, we consider the NetFlow 5-tuple flow definition: source IP, destination IP, source port, destination port, and transport protocol type. Two main tasks compose the traffic behavior prediction. The firs separates the sampled data into files representing 1-day collection each to train the models. The second calculates the entropy of a 5-tuple. The entropy is calculated for each 1-hour time window. The flow processing is divided into three main stages: (1) Shannon Entropy, (2) Wavelet Transform, and (3) Prediction Models.

1) Shannon Entropy: The information entropy, also known as Shannon Entropy, is a measure to analyze the uncertainty degree or concentration of information distribution. Originally, entropy was described as a measure on thermodynamics systems, but Claude Shannon extended it to the information theory in 1948. The concept applies to network traffic prediction because traffic has characteristics that are essentially random, such as the destination IP and source port. The entropy is mathematically expressed as

$$H(X) = -\sum_{i=1}^{n}(p_i)\log(p_i), \tag{1}$$

in which p_i is the probability of i-th result for a variable x.

2) Wavelet Transform: The wavelet function decomposes signals in the frequency domain and is useful for signal processing in the time domain. Chang *et al.* state that the function is an effective time-frequency analysis method after Fourier analysis [16]. There are two main types of wavelet transform, the Discrete Wavelet Transform (DWT), and Continuous Wavelet Transform (CWT).

Wavelet decomposition extracts low-frequency and high-frequency components because they satisfactorily produce a local analysis of the time series in the time and frequency domains. The components are the Component Detailed (CD) and the Component Approximate (CA). CD is responsible for generating linear components, whereas CA generates non-linear components. The extracted components may have information that makes predictions more accurate. Some prediction models have singularities that make them more efficient, depending on how the signal is processed.

DWT commonly provides a quick tool to remove noise from a signal. Considering a limited number of coefficients of the DWT components, it is possible to perform an inverse transform, obtaining a signal with reduced noise. The technique is useful in analyzing network traffic because, from the point of view of anomalous detection, the noise can represent an attack or background traffic.

3) Prediction Models: Auto-Regressive (AR), Moving Averages (MA), Auto-Regressive Moving Averages (ARMA), and Auto-Regressive Integrated Moving Average (ARIMA) models are examples of prediction algorithms based on time series analysis. The AR model is often used in stationary scenarios, which means that the scenario has a constant average over time. It is defined as AR*(p)*, which indicates a regressive model of order p, as represented in Equation 2. The MA model is used for smoothing and filtering the noise present in time series. There are several known variations, and the most important ones are Simple Moving Average (SMA), Weighted Moving Average (WMA), and Exponential Moving Average (EMA). The MA*(q)* model is defined in Equation 3. [17] The composition of the two models, AR and MA, in scenarios where there are auto-regressive (AR) and moving averages (MA) characteristics, generates the ARMA*(p,q)* model.

$$X_t = c + \sum_{i=1}^{p}\varphi_i X_{t\_i} + \varepsilon_t \tag{2}$$

$$Y_t = X_t - \sum_{i=1}^{q} -\varphi_i X_{t\_i} \tag{3}$$

The models well fit stationary time series. However, in several cases, the time series' are not stationary, and the ARIMA model is preferable. The model is ARIMA*(p,d,q)*, in which the parameters p, d, and q represent the order of the auto-regressive model, the number of necessary differentiation operations for a series to become stationary, and the order of the moving averages model, respectively. The variations of the AR model previously described aims to locate short-range dependencies, being ideal for the detection of anomalies, as it is necessary to act quickly to guarantee the accurate functioning of the network [18].

Another prediction method applied to time series relies on Recurrent Neural Network (RNN). The main principle of this method is tracking the dependencies of input values as a time series. Long short-term memory (LSTM) is a technique useful for anomaly detection, Intrusion detection system (IDS), or other signals processing applications, such as speech recognition. The basic structure of the LSTM is composed of a node, an input gate, an output gate, and a forget gate [19]. Input gate controls whether the memory cell updates, the forget gate controls whether the memory cell resets, and output controls whether the current cell state's information is visible.

Our proposal considers the entropy of network traffic features as an input time series. The proposed method applies the Discrete Wavelet Transform to retrieve both linear and non-linear components of the time series. For each component, we apply prediction using both ARIMA and LSTM. The prediction methods provide the forecasting for each signal component. Thus, we evaluate the Root Mean Square Error (RMSE) for each component and prediction method to identify the best prediction method suited to each component of every network traffic feature. The RMSE [17] is formalized in Equation 4. The final forecasting of each feature is the composition of the linear component's best prediction and the best prediction of the non-linear component.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(prediction_i - actual_i)^2}. \tag{4}$$

V. Results and Discussion

We assess the proposed approach using real network traffic statistics collected from the institutional wireless network of the *Universidade Federal Fluminense* using NetFlow protocol [7]. The network accounts for an infrastructure of more than 500 access points, peaking more than 5,000 users simultaneously associated in the network, and serves the entire academic community composed of more than 65,000 students, teachers, employees, and visitors.

The dataset[1] contains one-week traffic from the wireless network of a single campus of the university, namely *Praia*

[1] Derived data supporting the findings of this study are available from the corresponding author on request.

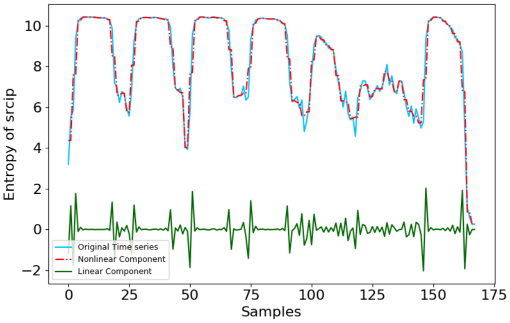


Fig. 1. Entropy of source IP feature and decomposition using DWT. Original series and linear and non-linear components extracted.

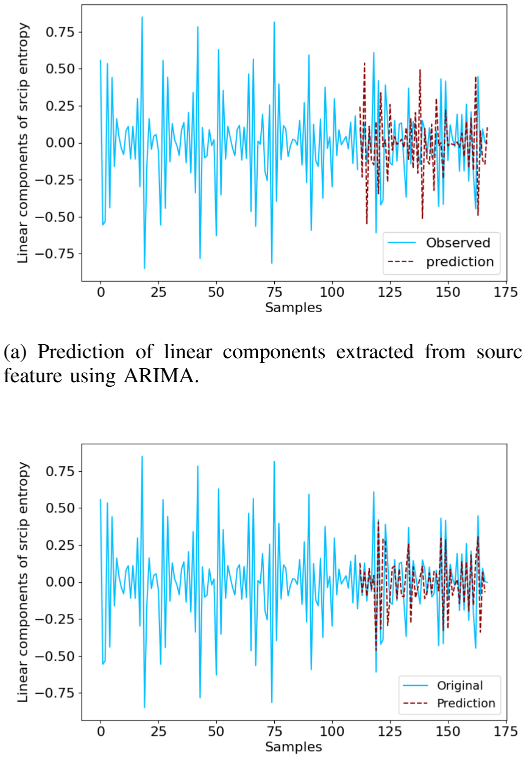
Vermelha. The data collection was performed between April 17-26, 2018, and generated a 16 GB size file. We use Python to process the data and to deploy prediction models on a computer equipped with Intel Quad Core i5-8265U processor with 1.60GHz of operation, 8GB RAM and 256 GB SSD disk storage, running Ubuntu 18.04 as operating system.

The evaluation's first step is to process the dataset to calculate Shannon Entropy. We calculate the entropy within 1-hour interval for each day. The total size of samples for the analyzed period is 168. Then, in the second step of the proposal, we apply the DWT using the Haar wavelet because it had a shorter run time compared to Daubechies wavelet an then we decompose the entropy time series into linear and non-linear components. Figure 1 shows the original time series for the source IP entropy compared with linear and non-linear reconstructed signal components after applying the inverse DWT. For the source IP, we highlight that the entropy values mainly present a non-linear behavior.

The last step of the proposal is to predict the entropy of features extracted from the network traffic. For this step, we train the prediction models with 67% of the input dataset and predict the following 33%. This split is equivalent to train over approximately 4-day usage, with a 1-hour entropy time window. Figure 2 shows the methods used to predict entropy based on linear components. On the other hand, Figure 3 presents a comparison between predictions using non-linear components. Both figures consider the source IP feature.

We use the RMSE metric calculated over LSTM and ARIMA models to compare both methods, as it is an efficient metric to validate the prediction models. RMSE represents measure of the average deviation between the observed and forecast values. When occurs a considerable difference between these values and squared is applied, this difference contributes to a high weight in the final prediction model error.Both scenarios show that LSTM outperforms ARIMA, as LSTM presents the lowest RMSE value, as shown in Tables I and II. However, the difference between the models for predicting the linear component is small, indicating that it is possible, depending on the sample's quantity, to use both.

We use a search method that performs tests for each value of p, d, and q, and chooses the best prediction parameter based on the Akaike Information Criterion (AIC). AIC is an estimator of out-of-sample prediction error and. Although we search the best parameters for ARIMA, it still performs worse than LSTM. Table I presents results for the ARIMA prediction. For each feature, we calculate the code execution time. It is noteworthy that for both the linear and non-linear cases, the parameters p, d, and q for the ARIMA model are the same for the features with the best results for training time and RMSE. In the linear model cases, the values 2, 1, 1 were stipulated to predict, while for the non-linear model, the values chosen by the estimator were 0, 1, 0.



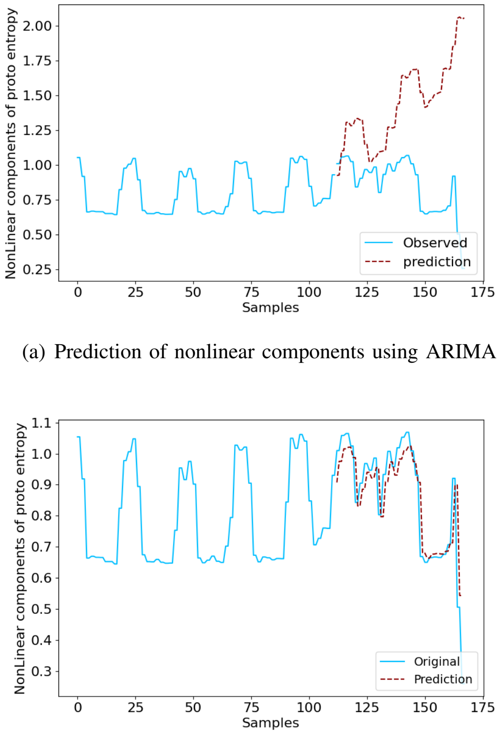
(a) Prediction of linear components extracted from source IP feature using ARIMA.

(b) Prediction of linear components extracted from source IP feature using LSTM.

Fig. 2. Comparison between models ARIMA and LSTM. (a) Prediction using ARIMA model for the linear components of source IP feature after the application of DWT. (b) Prediction using LSTM neural network for the linear components of source IP feature after the application of DWT. Both models perform well and follow the behavior of the signal, but LSTM follows the behavior of the original signal better.

Regarding the prediction with LSTM, Table II shows each feature's execution time, the RMSE value, and the maximum loss of each model epoch. It is possible to verify that the code execution time was significantly better than ARIMA and that the RMSE values also had a significant advantage. The linear component for the destination port feature (*dstport*) shows the best results of execution time and global RMSE. Protocol type feature presents the best performance for the RMSE metric, considering the non-linear components.

The results of RMSE for both LSTM and ARIMA predictions are shown in Figure 4. The RMSE for the linear components are shown Figure 4(a) and non-linear components are shown in Figure 4(b). It is possible to verify that the LSTM model outperforms ARIMA in both scenarios with a large difference, mainly on non-linear prediction. Indeed, it is expected



(a) Prediction of nonlinear components using ARIMA

(b) Prediction of nonlinear components using LSTM

Fig. 3. Comparison between models ARIMA and LSTM. (a) Prediction using ARIMA model for non-linear components of source IP feature after applying DWT. (b) Prediction using LSTM neural network for non-linear components of source IP feature after the application of DWT. The ARIMA model diverges from the original signal and the error tends to increase over time.

TABLE I
COMPARISON BETWEEN LINEAR AND NON-LINEAR ARIMA PREDICTION

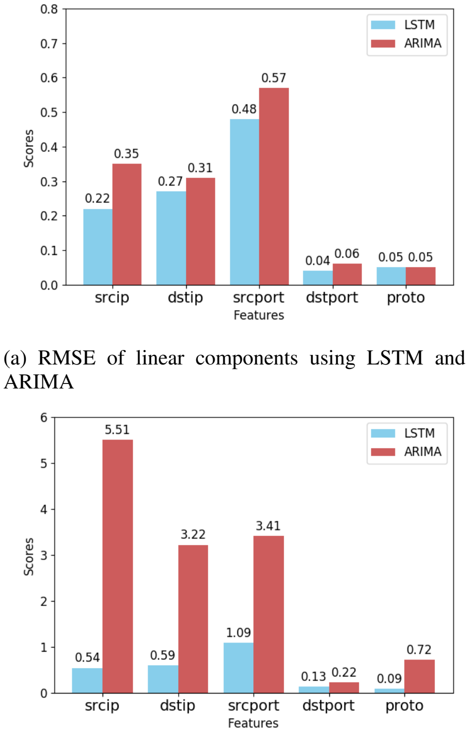
	Runtime (seconds)	RMSE	AIC	Model
srcip (linear)	**38s**	0.35	-55.07	2,1,1
dstip (linear)	42s	0.31	-40.0	3,1,1
srcport (linear)	40s	0.57	42.30	3,1,1
dstport (linear)	44s	0.06	-183.66	2,1,1
proto (linear)	44s	**0.05**	-408.29	2,1,1
srcip (nonlinear)	79s	5.51	43.23	0,1,0
dstip (nonlinear)	45s	3.22	71.23	1,1,1
srcport (nonlinear)	39s	3.41	175.36	1,1,1
dstport (nonlinear)	55s	**0.22**	-166.62	0,1,0
proto (nonlinear)	**33s**	0.72	-356.17	0,1,0

that LSTM outperforms ARIMA in forecasting the non-linear component because of the high adaptability capacity of the neural networks to deal with non-linear problems. However, the ARIMA model fails to provide better prediction even to linear components for all considered features. We emphasize that the best time performance for the neural network-based models is due to the algorithms' optimized implementation within the Keras library[2].

[2] Available at https://keras.io/.

TABLE II
COMPARISON BETWEEN LINEAR AND NON-LINEAR LSTM PREDICTION

	Runtime (seconds)	RMSE	Maximum Loss
srcip (linear)	13.98s	0.22	0.2037
dstip (linear)	12.93s	0.27	**0.1638**
srcport (linear)	12.94s	0.48	0.2159
dstport (linear)	**12.82s**	**0.04**	0.2360
proto (linear)	13.56s	0.05	0.2252
srcip (nonlinear)	13.73s	0.54	0.4240
dstip (nonlinear)	13.17s	0.59	0.6059
srcport (nonlinear)	13.29s	1.09	0.5938
dstport (nonlinear)	**13.13s**	0.13	0.3812
proto (nonlinear)	13.26s	**0.09**	**0.1951**

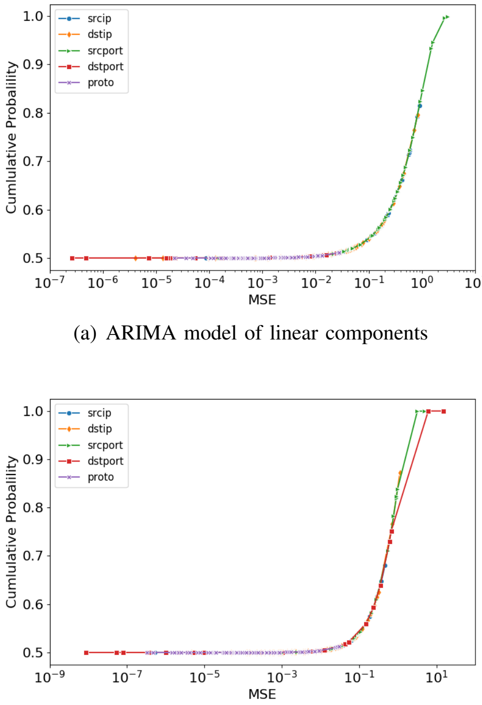


(a) RMSE of linear components using LSTM and ARIMA

(b) RMSE of non-linear components using LSTM and ARIMA

Fig. 4. Comparison of RMSE from linear and non-linear components. a) For linear components, LSTM outperforms ARIMA, but the RMSE values are almost equal for both methods applied to all features. b) The RMSE values for LSTM are significantly lower than ARIMA's values.

We calculate a Cumulative Distribution Function (CDF) to compare the probability of the analyzed features to reach a given Mean Square Error (MSE). The MSE is another measure of the quality of prediction models, in which lower values indicate higher accuracy of predictions. Figure 5 show the CDF of MSE values for each prediction model. The linear component of the destination port feature using LSTM presents the lowest predicting error due to this feature's low values of entropy, indicating that a variation in this parameter can be an anomaly.



(a) ARIMA model of linear components

(b) LSTM model of linear components

Fig. 5. Cumulative probability (CDF) of MSE for each model. (a) CDF for ARIMA of linear components, (b) CDF for LSTM model of linear components. ARIMA error uperbound is lower than LSTM uperbound, indicating that the maximum errors on the ARIMA predictions for the linear component are lower than these achieved by LSTM.

VI. Conclusion

The entropy calculation aims to demonstrate the degree of randomness that exists in a given system. It proves to be useful in predicting anomalies and network traffic. Even having seasonal characteristics, depending on the usage, network traffic presents high dispersion in the statistics related to IP stack flow features, such as source and destination IP or source and destination ports. The ARIMA statistical model analyzes the temporal classifications to understand the historical data and forecasts traffic based on moving averages and linear regression. In contrast, the LSTM neural network model has the advantage of being more efficient in processing than ARIMA. Our results show that the LSTM neural network provided low error in predicting both linear and non-linear components of the entropy time series, while ARIMA provided a lower upper bound to the error on the linear components compared to LSTM.We draw our conclusion that a hybrid method that considers both ARIMA and LSTM is the best solution for predicting the abnormal behavior of a large scale network, for two reasons, first because both uses low computational resources and second, because for non-linear components the LSTM demonstrated superior performance in contrast with ARIMA, making possible to forecasting one-step ahead since the predominant traffic is non-linear, while for linear components ARIMA is best choice to make predictions for long-term for non-stationary linear processes. As future work, we intend to apply a one-class support vector machine classifier to detect anomalies in predicted future samples of a time series for real time analysis.

References

[1] Cisco, *Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper*. Cisco Systems, 2019.

[2] D. M. F. Mattos, P. B. Velloso, and O. C. M. B. Duarte, "An agile and effective network function virtualization infrastructure for the internet of things," *Journal of Internet Services and Applications*, vol. 10, no. 1, p. 6, Mar 2019.

[3] D. S. V. Medeiros, H. N. Cunha Neto, M. A. Lopez, L. C. S. Magalhães, N. C. Fernandes, A. B. Vieira, E. F. Silva, and D. M. F. Mattos, "A survey on data analysis on large-scale wireless networks: online stream processing, trends, and challenges," *Journal of Internet Services and Applications*, vol. 11, no. 1, p. 6, Oct 2020.

[4] D. M. F. Mattos, O. C. M. B. Duarte, and G. Pujolle, "Profiling software defined networks for dynamic distributed-controller provisioning," in *2016 7th International Conference on the Network of the Future (NOF)*, 2016, pp. 1–5.

[5] C. Brenda Zerbini, L.Fernando Carvalho, T. Abrão, and M. Lemes Proença Jr, "Wavelet against random forest for anomaly mitigation in software-defined networking," in *Applied Soft Computing Journal*. Elsevier, 2019.

[6] R. Madan and P. S. Mangipudi, "Predicting computer network traffic: A time series forecasting approach using dwt, arima and rnn," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1–5.

[7] Reis, Lucio Henrik A, Magalhães, Luiz Claudio S., de Medeiros, Dianne Scherly V., and Mattos, Diogo M. F., "An unsupervised approach to infer quality of service for large-scale wireless networking," *Journal of Network and Systems Management*, vol. 28, pp. 1228–1247, 2020.

[8] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," in *Computer Networks*. Elsevier, 2014.

[9] Vaclav Bartos, Martin Zadnik, Sheikh Mahbub Habib, and Emmanouil Vasilomanolakis, "Network entity characterization and attack prediction," in *Future Generation Computer Systems*. Elsevier, 2019.

[10] L. Wang, Q. Li, Y. Jiang, X. Jia, and J. Wu, "Woodpecker: Detecting and mitigating link-flooding attacks via sdn," in *Computer Networks*. Elsevier, 2018.

[11] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, and M. S. Haghighi, "Anomaly detection in automated vehicles using multistage attention-based convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[12] P. Holgado, V. A. Villagrá, and L. Vázquez, "Real-time multistep attack prediction based on hidden markov models," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 134–147, 2020.

[13] C. Yang, "Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment," *Cluster Computing*, vol. 22, no. 22, pp. 8319–8317, 7 2019.

[14] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 187–193.

[15] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li, "Traffic flow prediction based on combination of support vector machine and data denoising schemes," *Physica A: Statistical Mechanics and its Applications*, vol. 534, p. 120642, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437119302262

[16] Zihan Chang, Yang Zhang, and Wenbo Chen, "Electricity price prediction based on hybrid model of adam optimized lstm neural network and wavelet transform," in *Energy*. Elsevier, 2019.

[17] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401.

[18] L. Nie, D. Jiang, S. Yu, and H. Song, "Network traffic prediction based on deep belief network in wireless mesh backbone networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2017, pp. 1–5.

[19] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.