Security-aware Data-driven Intelligent Transportation Systems

Malik, Jahanzaib; Akhunzada, Adnan; Bibi, Iram; Talha, Muhammad; Jan, Mian Ahmad; Usman, Muhammad

Published in:
IEEE Sensors Journal

Link to article, DOI:
[10.1109/JSEN.2020.3012046](https://doi.org/10.1109/JSEN.2020.3012046)

Publication date:
2021

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Malik, J., Akhunzada, A., Bibi, I., Talha, M., Jan, M. A., & Usman, M. (2021). Security-aware Data-driven Intelligent Transportation Systems. *IEEE Sensors Journal*, 21(14), 15859-15866.
<https://doi.org/10.1109/JSEN.2020.3012046>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Security-aware Data-driven Intelligent Transportation Systems

Jahanzaib Malik, Adnan Akhunzada†, Iram Bibi, Muhammad Talha, Mian Ahmad Jan*, and Muhammad Usman*

Abstract— Intelligent transportation systems have been envisioned to bring more intelligence and cooperative sensing to meet the imminent demands of overall improved autonomous transportation. However, dynamic era of modern applications and fixed architecture of legacy Internet needs flexible, innovative, adaptive, and programmable software defined intelligent transportation systems (SD-ITS). The centralized control intelligence of SD-ITS can be a potential primary target of the prevalent cyber threats and attacks that can simply throw the entire network into chaos. The authors propose a DL-driven multi-vector scalable attack detection framework leveraging graphical processing unit (GPU) empowered Bidirectional Long Short-Term Memory (BLSTM) to efficiently tackle exponentially growing diverse sophisticated attacks that primarily target the control unit of the SD-ITS. The proposed technique has been rigorously evaluated with current state-of-the-art publicly available Flow-based dataset (i.e., CICIDS2017) using standard performance metrics. Further, the proposed mechanism is compared with contemporary benchmarks (i.e., DL algorithms). Extensive experimental results exhibit out-performance of the proposed technique in term of detection accuracy with a trivial trade-off computational complexity. Finally, the study also employed 10-fold cross validation to explicitly show unbiased results.

Index Terms—Deep learning, software defined intelligent transportation systems (SD-ITS) cyber threats & attacks, data-driven intelligent transportation systems

I. INTRODUCTION

SOFTWARE defined network has emerged as a promising next generation networking paradigm. Flexible, dynamic, and adaptive architecture of SDN has changed the way of traditional Internet multifaceted network design [1], [2]. SDN decouples the traditional hierarchical combination of network control by empowering the control plane to be intelligent, programmable, innovative and centralized decision maker [3]. The control plane also provides a complete abstraction of the underlying software defined intelligent transportation systems (SD-ITS). Despite the promising control plane of the SD-ITS that manage decisions and logic related to sensing and network traffic processing also viewed as a single point of failure [4]. Thus, the centralized control intelligence of SD-ITS is primarily a potential target of various sophisticated

threats and attacks to disrupt the entire underlying data driven intelligent transportation systems [5], [6].

Nevertheless, the literature is evident that SDN-based networks are confronted with varied drastic subtle attacks like Denial of service (DoS), Distributed denial of service (DDoS), Brute force, Web attacks, Botnet etc [7]. Consequently, the security of the SDN controller is of paramount concern that deserves to receive a decent expanse of attention [8], [9]. To detect and mitigate the evolving latest cyber security threats, intelligent learning-based detection mechanisms should be devised. Intrusion detection system (IDS) is designed to handle anomalies, and malevolent behaviours [10]. However, conventional signature-based IDS always need a set of knowledge branded as signatures or patterns of potential attacks that should be regularly updated for a timely detection. Additionally, it fails to identify zero-day attacks. Thus, the current trend is to explore intelligent DL-based anomaly detection systems having capability to identify novel and zero-day attacks. DL-based algorithms have built in ability of key feature extraction to identify dynamic prevalent cyber-threats more effectively [11].

Despite leveraging high-performance, unprecedented automation and flexibility of next generation SD-ITS, security is still an arresting challenge that reduces the significant growth of promising SDN-enabled emerging networks [12]. Moreover, launching cyber security attacks against these emerging networks have devastating impact than traditional networks. Since the control plane of the SDN can extend many potential networks at the data plane such as Software Defined Intelligent Transportation System (SD-ITS), Internet-of-Things (IoTs) [13], [14], Fog-to-Things (FoTs) [15], [16], Edge-of-Things (EoTs) [17], Industrial Internet-of-Things (IIoTs) [18] and vehicular networks [19] etc. Therefore, the control plane is having huge potential. On the other hand, the centralized control plane is also being recognized as a single point of failure and a compelling target to throw the entire underlying network into chaos.

The main objective of the paper is to design a highly scalable and efficient intrusion detection framework to protect the control intelligence of the SD-ITS from varied drastic multi-vector cyber threats and attacks (i.e., FTP-Patator, SSH-Patator, DoS-Goldeneye, DoS-Hulk, Slow Httptest, DoS-Slowloris, Heartbleed, XSS, SQL Injection, Brute force, Infiltration, Botnet, DDoS, Port scan). The key contributions of the paper are manifold.

- To the best of our knowledge, this is perhaps the first study that comprehensively evaluate multi-vector attacks and evolving cyber threats in SD-ITS using current state-of-the-art Flow-based refined publicly available dataset

† denotes equal-first author

* denotes corresponding authors

Jahanzaib Malik and Iram Bibi is with COMSATS University, Islamabad, Pakistan (e-mail: jahanzaib.int@gmail.com, ask4iram@gmail.com).

Adnan Akhunzada is with Comsats University Islamabad and DTU Compute, Technical University of Denmark (e-mail: adnak@dtu.dk).

Muhammad Talha is with Deanship of Scientific Research, King Saud University Riyadh, Saudi Arabia (e-mail: mnaseem@ksu.edu.sa).

Mian Ahmad Jan is with the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan (e-mail: mianjan@awkum.edu.pk)

Muhammad Usman is with School of Science, Engineering and Information Technology, Federation University, Melbourne, Australia (e-mail: m.usman@federation.edu.au)

(i.e., CICIDS2017) leveraging CUDA-enabled Bidirectional Long Short-Term Memory (BLSTM). The proposed mechanism is highly scalable and effective.

- The authors rigorously evaluate the proposed mechanism using standard and extended performance metrics (e.g., Accuracy, Precision, Recall, F1-Score, AU-ROC, etc.).
- For verification purpose, the proposed mechanism is compared with current benchmark DL-driven algorithms (i.e., Cuda enabled convolutional neural network (Cu-CNN), Cuda enabled deep neural network (Cu-DNN)). We have also compared the proposed mechanism with current state-of-the-art. Our proposed technique outperforms in terms of detection accuracy with a trivial trade-off computational complexity.
- Finally, to show that the results are completely unbiased, 10-fold cross validation has been performed.

The remaining paper is organized as follows. Section II represents related work. Section III gives a comprehensive overview of our proposed system, including system design, dataset, a detailed account of its various components and algorithms description. Section IV presents experimental results & discussion which includes evaluation metrics and discussion on achieved results. Finally, Section V concludes the paper with future remarks.

II. RELATED WORK

In recent years, researchers have proposed varied intrusion detection mechanisms to protect against malicious threats and attacks in software defined based networks. Most of the existing works are machine learning based anomaly detection mechanisms [20]. However, deep learning (DL) architectures (e.g., Convolutional Neural Network (CNN), Deep Belief Network (DBN), Recurrent Neural Networks (RNN), Deep Neural Network (DNN), Ensemble and Hybrid DL models are still in its infancy towards comprehensive evaluation of diversified attacks in SD-ITS using Flow-based publicly available datasets.

Li et al. in [21], propose a LSTM based SQL injection attack detection scheme for intelligent transportation system. The proposed technique is capable to learn representation of data automatically and also defines the sample generation method. The dataset includes with positive collected samples of SQL injection from exploit-db and WooYun vulnerability submission platform. The LSTM-based network achieves 93.47% detection accuracy. Anomaly detection framework for DDoS is proposed in [22] for intelligent transportation system. For traffic classification, Random Forest (RF) is adopted and evaluated on NSL-KDD and UNSW-NB15. The experimental results show the performance of proposed detection algorithm with accuracy rate of 99.95% and 98.75% for binary classification. The authors in [23], presents a consolidated framework for DDoS attack detection in SDN using ensemble Restricted Boltzmann Machine (RBM) with Support Vector Machine (SVM) classifiers. KDD 99 and Carnegie Mellon University (CMU) based insider threat datasets have been utilized for evaluation that achieves 99.98% detection accuracy. In [24], the authors contribute by proffer an efficient intrusion

detection system deploying Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) in SDN. The proposed work is evaluated both on NSL-KDD and CICIDS2017 dataset that achieves an accuracy of 89% and 99% respectively. Similarly, in [25], the authors construct a deep learning-based detection framework for DDoS attack detection using Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), and Convolution Neural Network (CNN). ISCX2012 dataset is utilized that results in 98% detection accuracy.

Moreover, this technique also carries prevention mechanism (i.e., implementing a drop policy to discard malicious network flows) at the SDN controller. In [26], devised a profound GRU-RNN enabled intrusion detection system in SDN that utilizes six basic features of the benchmark NSL-KDD dataset. The algorithm achieves 89% detection accuracy. Authors in [27], use DL-based algorithms (i.e., Multilayer Layer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) with four different adversarial testing schemes (i.e., Fast Gradient Sign Method (FGSM), JSMA reverse (JSMA-RE) and Jacobian-based Saliency Map Attack (JSMA)) for adversarial attack detection. The data for the experimentation has been collected through Wireshark and tshark tools. The proposed scheme attains 98% detection accuracy. A flow-based anomaly detection mechanism has been devised for TCP, UDP and ICMP attacks using Deep Neural Network (DNN) in [28]. SNORT with DNN is used to acquire 85% identification accuracy. The authors in [29], present Recurrent Neural Network (RNN) trained on NSL-KDD dataset for network intrusion detection. This model is evaluated for binary and multi attack classification that achieves detection accuracies (i.e., 83.28%, 97.09%) respectively. A DDoS identification framework in [30], consolidates Stacked Auto-Encoder (SAE) based DL approach in SDN. The performance of proposed study on network traffic gathered from home wireless network that comprises of typical Internet traffic with varied DDoS attack traffic and gets 95.65% detection accuracy.

To the best of our knowledge, this is perhaps the first effort that thoroughly studies and evaluate fifteen diverse attacks using current state-of-the-art Flow-based refined publicly available dataset (i.e., CICIDS2017) leveraging CUDA-enabled BLSTM. The existing literature largely neglects evaluating ML and DL-based intrusion detection systems (IDS) with dataset containing pure flow-based features in SD-enabled intelligent transportation systems (SD-ITS). Generally, few Standard performance metrics have been employed to evaluate the existing IDS architectures. However, this study employs most of the standard performance metrics for a comprehensive evaluation. Further, speed efficiency of the proposed mechanism has also been evaluated that seems rare in some of the existing literature. Finally, to show explicitly unbiased results, a 10-fold cross validation has also been performed.

III. METHODOLOGY

This section presents complete methodology of the proposed system that includes the complete description of proposed framework, dataset description, preprocessing of dataset (i.e., Handling Missing Data, Handling Categorical Data, Fea-

Fig. 1. The Simplified overview of SD-ITS.

ture Scaling) and architectural description of utilized DL-Algorithms.

A. Proposed Framework

Software Defined Network (SDN) emerged as promising cutting-edge networking paradigm. The architectural engineering of SDN is depicted by the separation of the control plane from the data plane [31]. The control plane, an intelligent centralized logical entity provides and abstract view of the underlying network. Moreover, the control plane is responsible for decision making, information handling and forwarding capabilities of the network [32]. Consequently, data plane is comprised of basic forwarding devices (i.e., routers and switches) and other SDN agents. SDN relies mainly upon the control plane since it is programmable and have the ability to scale and integrate diverse modules [33]. Consequently, the authors have executed their detection module at control plane. For the detailed and comprehensive overview of the SDN architecture, please explore the authors published work [31], [33].

Fig.1 shows a simplified overview of the SD-ITS architecture also depicting our proposed framework at the control plane. The SD-ITS architecture comprises of three planes (i.e., application plane, control plane, and data plane). The power of SD-ITS lies at the control plane, responsible for the

whole centralized control intelligence and network management. Moreover, the control plane is programmable, adaptive and innovative. Southbound APIs (Application Programming Interfaces) have been designed to facilitate the communication between the SDN Controller and the underlying data plane network through SDN soft switches. The complete overview of the SD-ITS architecture is detailed in authors published work [31]. Further, our implemented DL-driven detection module can be customized and work as an extended module of any commercial SDN controller such as Floodlight, OpenDaylight, POX, etc. That is the main reason that our proposed work is highly scalable. Finally, the framework is capable to efficiently identify a verity of cyber threats and attacks well-timed and effectively.

B. Dataset

To overthrow the existing gap of being outdated dataset for network intrusion detection in SDNs, CICIDS2017 has been generated by Canadian institute of Cybersecurity [34]. CICIDS2017 dataset contains normal and potential attacks signatures, resembles with genuine certifiable information (PCAPs). The dataset comprises of system traffic together with utilization of CICFlowMeter for labelled streams dependent on the time stamp, source and destination IPs, source and destination ports, protocols and label. The executed attacks incorporate FTP-Patator, SSH-Patator, DoS-Goldeneye, DoS-Hulk, DoS-Slowhttptest, DoS-Slowloris, Heartbleed, XSS, SQL Injection, Brute force, Infiltration, Botnet, DDoS, Port Scan. The dataset contained 85 features with 15 unique class labels, one normal and fourteen diverse attack labels aforementioned. The authors have realized and discovered that attributes *Fwd_Header_Length* appeared twice in the list of attributes. Moreover, the '*Flow-ID*' contains 5 different features (i.e., source and destination IPs, source and destination ports, protocols) that already appeared as distinct identities. So, these 2 instances (i.e., *Flow-ID*, *Fwd_Header_Length*) have been removed from dataset. Now the dataset contains 83 distinctive feature values. Detailed description of CICIDS2018 dataset could be found in [34].

C. Pre-Processing

In the adopted research methodology, initial phase can be demonstrated as preprocessing of data that includes handling missing data, categorical data and feature scaling of dataset.

1) Handling Missing Data

By cavernous observation of CICIDS2017 dataset, it satisfies every single criterion such as complete labelled traffic with diverse network configuration, attacks diversity and heterogeneity but in the meantime, dataset likewise contains shortcoming of scattered presence and missing values. For scaled-down version of dataset, selected 529918 benign records combined with 559613 records of diverse attacks signatures. Additionally, the dataset has issue of missing data and presence of nan values. Designed python preprocessing script has been used to expel missing values instances as well as instances with nan values. The dataset has been normalized having

no redundant values. The complete description after traffic distribution and normalization from 8 different files can be seen in [34].

2) Handling Categorical Data

In CICIDS2017 dataset, 80 features among 83 have numeric values while the remaining are nominal. For feeding data to an algorithm in understandable format, the conversion of non-numeric values i.e., *Sender_IP*, *Destination_IP* and *Timestamp* into numeric values have been performed. Further, a function of socket programming is utilized to convert IP addresses to integers, and for the conversion of timestamp the authors simply converted the DateTime to integer. This study also label encode each target class to get unique integer value and expressed them in OHE (One hot Encoded) format for training.

3) Feature Scaling

The dataset contains features having huge difference between their maximum and minimum values. For stable convergence of weights, biases and to avoid gradient vanishing and exploding, the dataset features has been normalized using *MinMaxScaler* to map values between 0 and 1.

D. DL-driven Detection Architecture

This study implemented three state-of-the-art cuda enabled DL-driven architectures (i.e., Cu-BLSTM, Cu-CNN, Cu-DNN). The complete design of GPU-accelerated BLSTM architectures *Layers*, *Neurons*, *Activation Functions*, *Loss Function*, *Optimizer*, *Batch Size* and *Epochs* in the experiment are BLSTM Layers, (300,350,300,125,50,16,15), ReLu, Categorical Cross-Entropy, Adam, 256 and 10 respectively. The working of proposed bidirectional LSTM with 10-fold cross validation for network intrusion detection system is also described in Algorithm 1.

E. Algorithms

The brief explanation of the proposed algorithm is defined to show the basic working of used architectures like Bidirectional Long Short-Term Memory (BLSTM).

Bidirectional Long Short-Term Memory

Recurrent neural network (RNN), a family of Artificial Neural Network (ANN) has great potential to learn from previous time-steps. In hidden layer of RNN, each node's output is given as a contribution to the concurrent node at each timestep. RNN continuously learn from past timesteps by Back propagation Through Time (BPTT). When time-steps cross, standard RNN cannot perform better. The back-propagation prompts vanishing or explode of error signals, prompting wavering weights, which makes the system execution poor. To resolve the vanishing gradient issue, analysts worked on Long-Short-Term-Memory (LSTM) that bridges the stripped time gaps. Researchers have revised LSTM to manage the shortcoming of the standard LSTM model: Bidirectional Long Short-Term-Memory (BLSTM). The detailed description of BLSTM could be seen in [35].

Algorithm 1 10 Fold for Bidirectional LSTM

Require: Dataset-Training T, Dataset-Testing t, LSTM-layers L, Dense-layer D, K-Folds K, Epochs E, Batch-Size B, Weights W

Ensure: K-Fold Cross Validation & Saving Results

```

1: function CROSS-VALIDATION()
2:   for Fold ← 1 to K do
3:     for Epoch ← 1 to E do
4:       for Sample ← 1 to T do
5:         Train w.r.t B from T
6:         Calculate Loss w.r.t B
7:         if Wrong Prediction then
8:           Update W
9:         end if
10:      end for
11:    end for
12:    Save Results for K
13:  end for
14: end function

```

Finally, the "output gate" settles on what data from the cell state is required to output. The output data is pass through the function known as tanh before going through sigmoid to ensure that all values are between -1 and +1.

Confusion matrix, without normalization

True label \ Predicted label	Normal	FTP-Patator	SSH-Patator	DOS-Goldeneye	DOS-Hulk	DOS-Slowhttptest	DOS-Slowloris	DOS-Heartbleed	XSS	SQL Injection	Brute-Force	Infiltration	Bot	DDOS	Port Scanning
Normal	52632	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FTP-Patator	3	772	0	0	0	0	0	0	0	0	0	0	0	0	0
SSH-Patator	0	1	611	0	0	0	0	0	0	0	0	0	0	0	0
DOS-Goldeneye	3	0	0	1066	6	0	0	0	0	0	0	0	0	0	0
DOS-Hulk	0	0	0	0	23015	0	0	0	0	0	0	0	0	0	0
DOS-Slowhttptest	1	0	0	1	0	555	0	0	0	0	0	0	0	0	0
DOS-Slowloris	2	1	0	1	1	32	576	0	0	0	0	0	0	0	0
DOS-Heartbleed	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
XSS	0	0	0	0	0	0	0	0	1	0	64	0	0	0	0
SQL Injection	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
Brute-Force	0	0	0	0	0	0	0	0	1	0	166	0	0	0	0
Infiltration	2	0	0	1	0	0	0	0	0	0	1	1	0	0	0
Bot	0	0	0	0	0	0	0	0	0	0	0	0	182	0	0
DDOS	0	0	0	0	0	0	0	0	0	0	0	0	2	12899	0
Port Scanning	0	0	0	0	0	0	0	0	0	0	2	0	0	3	15997

Fig. 2. Confusion matrix for proposed Cu-BLSTM

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results and discussion section describe the details of experimental setup for proposed framework, basic standard performance evaluation metrics and brief discussion on achieved results.

A. Experimental Setup

To get maximum performance from proposed intrusion detection architecture for SD-ITS, the paper has implemented framework using high level Keras library. The experiment is executed in an environment of Intel Core i7 - 7700HQ with 16GB RAM. The computational power of graphical processing unit (GPU) is also utilized to enhance for training and testing. We used NVIDIA GeForce GTX 1050Ti for our experimentation.

TABLE I
ACCURACY BY CLASS

CLASS	Cu-BSTLM	Cu-CNN	Cu-DNN
Benign	99.99	99.98	99.98
Botnet	99.9	99.4	79.02
BruteForce	99.99	100	100
DDOS	99.98	99.97	99.98
DoS-GoldenEye	99.2	98.93	98.34
DoS-Heartbleed	0	0	0
DoS-Hulk	99.99	99.95	100
DoS-Slowhttptest	99.6	99.09	92.69
Dos-Slowloris	99.3	98.10	90.81
FTP-Patator	99.97	99.87	99.87
Infiltration	75	75	0
Port Scanning	99.97	99.94	99.89
SQL-Injection	0	0	0
SSH-Patator	99.3	99.32	99.50
XSS	6.1	1.53	0
Total Accuracy	99.89	99.83	99.81

B. Evaluation Metrics

An ideal IDS system entails high detection accuracy rate of anomalous behaviour with very low false alarms. For performance evaluation of proposed network intrusion detection system for SD-based intelligent transportation systems (SD-ITS), the study have employed standard evaluation metrics that are Detection Accuracy Precision, Recall, ROC, F1-Score, Testing time, True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), False Discovery Rate (FDR), Negative Predictive Value (NPV), False Omission Rate (FOR), Matthews Correlation Coefficient (MCC) and Bookmaker Informedness (BM). True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are being noticed from confusion matrix [36].

Accuracy

Numbers of correctly labelled records verses total number of records. It is the most well-known evaluation parameter for the estimating classifier

Precision

Precision shows the proportion of intrusion anticipated by a network intrusion detection system as a real intrusion. As the value of P is higher, then the probability of lower false alarm rate is. Precision is also well-known as the Positive Predictive Value (PPV).

Recall

Number of records predicted correctly division by total positive records for each class. The objective for recall is to achieve high value for it.

F1-score

F1-Score/F-Measure presents harmonic mean between recall and precision. F1-Score demonstrate the balance between PPV and Recall.

AU-ROC Curve

Plotting TP rate and FP rate in two dimensions will plot a curve known as area under curve. AU-ROC is the overall performance of system, more area under the curve higher the performance of the model.

C. Discussion

The CUDA-enabled DL-driven framework is designed for 15 distinct classes that incorporates benign and 14 pervasive lethal attacks. For performance evaluation, this paper compares our proposed CUDA-Empowered Bidirectional Long Short-Term Memory (BLSTM) technique with our other constructed benchmarks like CUDA-enabled Convolutional Neural Network (CNN) and Deep Neural Network (DNN). Standard performance evaluation metrics have been used to rigorously evaluate the proposed architecture. Further, 10-fold cross validation has also been performed to show explicitly unbiased results.

Confusion Matrix

Fig.2 illustrates the confusion matrix of the proposed Cu-BLSTM in a multi attack scenario to evaluate the performance. The confusion matrix of Cu-BLSTM outlines that 2 attack classes (i.e., Heartbleed, SQL Injection) are not properly learned by the proposed algorithm because of smaller number of instances in training phase. Though, the confusion matrix clearly shows the excellent performance of the proposed algorithm.

10-Fold Cross Validation

To show explicitly unbiased results, the authors have applied the 10-fold cross validation technique for intrusion detection. For standard evaluation metrics (i.e., Detection Accuracy, Precision, Recall, F1-Score, TNR, MCC and so forth) the standard average has been calculated to represent achieved results efficiently.

Per-Class Accuracy

The per-class testing accuracy of 15 varied classes have also been calculated to show ultimate performance of proposed framework. Table I clearly depicts that all the three classifiers have learned benign traffic signatures outstandingly. However, when it comes to SQL Injection, XSS, Heartbleed and Infiltration the proposed Cu-BLSTM largely neglected to learn SQL Injection and Heartbleed. The reason behind is obvious that the dataset carries very few instances of these attacks. Additionally, the graph approaches zero, when Cu-CNN and Cu-DNN tried to learn 4 attack classes (i.e., Heartbleed, XSS, SQL Injection and Infiltration). For rest of the classes, CUDA-enabled DL models performed well while achieving over 90% detection accuracy. Finally, the per class graph also demonstrates the efficient performance of the proposed algorithm even in terms of the largely neglected learned classes of attacks.

Accuracy, Precision, Recall and F1-Score

The results of each fold for the referenced metrics (i.e., Detection Accuracy, Precision, Recall and F1-Score) can be

TABLE II
10-FOLD CROSS VALIDATION OF ACCURACY, PRECISION, RECALL AND F1-SCORE FOR GPU-ACCELERATED BLSTM, CNN AND DNN

Folds	Accuracy (%)			Precision (%)			Recall (%)			F1-Score (%)		
	BLSTM	CNN	DNN	BLSTM	CNN	DNN	BLSTM	CNN	DNN	BLSTM	CNN	DNN
CUDA-ENABLED												
1	99.903	99.864	99.837	99.807	99.847	99.807	99.969	99.875	99.858	99.969	99.875	99.858
2	99.905	99.830	99.828	99.755	99.824	99.755	99.962	99.828	99.894	99.962	99.828	99.894
3	99.886	99.822	99.855	99.802	99.832	99.802	99.954	99.803	99.902	99.954	99.803	99.902
4	99.905	99.636	99.838	99.811	99.365	99.811	99.952	99.892	99.858	99.952	99.892	99.858
5	99.914	99.855	99.863	99.890	99.815	99.890	99.973	99.888	99.830	99.973	99.888	99.830
6	99.893	99.843	99.846	99.863	99.762	99.863	99.950	99.916	99.820	99.950	99.916	99.820
7	99.912	99.823	99.790	99.768	99.695	99.768	99.964	99.943	99.799	99.964	99.943	99.799
8	99.899	99.854	99.778	99.771	99.805	99.771	99.932	99.896	99.773	99.932	99.896	99.773
9	99.875	99.789	99.840	99.857	99.618	99.757	99.834	99.950	99.818	99.834	99.950	99.818
10	99.894	99.867	99.825	99.783	99.839	99.783	99.939	99.888	99.860	99.939	99.888	99.860

seen in Table II. The Table clearly delineates that our proposed Cu-BLSTM achieved excel performance with 99.89% detection accuracy, 99.84% precision, 99.94% recall and 99.93% F1-score compared to Cu-CNN and Cu-DNN.

Comparison with the Current State-of-the-Art

Furthermore, the comparison of the proposed intrusion detection system against the present best in class models is likewise outlined in Table III. It is clearly evident from the results that the proposed model beats the current state of the art schemes.

Fig. 3. TNR, MCC, BM and NPV Rate for Cu-BLSTM, Cu-CNN and Cu-DNN

TNR, MCC, BM and NPV

To broaden the circle of assessment for three different GPU-Accelerated benchmark model's, TNR, MCC, BM and NPV values have also been drawn from confusion matrix. True Negative Rate (TNR) is the ratio of the correctly classified attacks samples w.r.t total attacks. While, Matthews Correlation Coefficient (MCC) metric represents the correlation between observed and predicted classification. Bookmarker Informativeness (BM) is used for the evaluation of overall discriminative power of a diagnostic procedure. However, Negative Predictive Value (NPV) measures the ratio of attacks samples that were classified correctly divided to the overall number of attacks

Fig. 4. AU-ROC Curve for proposed Cu-BLSTM

predicted samples. An algorithm with high accomplished values of TNR, MCC, BM, NPV is considered as a proficient anomaly detection mechanism. Fig.3 is depicting the values of TNR, MCC, BM and NPV as 99.8%, 99.88%, 99.6% and 96% for Cu-BLSTM respectively.

FNR, FPR, FDR and FOR

The False Negative Rate (FNR) is basically extent of true samples that were not classified correctly. While, False Positive Rate (FPR), shows the proportion between the misclassified false samples to the overall false samples. Furthermore, False Omission Rate (FOR) and False Discovery Rate (FDR) basically calculates complements of the NPV and PPV respectively. The model of Cu-BLSTM achieved FNR, FPR, FDR and FOR of 0.04%, 0.15%, 0.16% and 0.04% as shown in Fig.5.

AU-ROC's

AU-ROC's for proposed DL algorithm (i.e., Cu-BLSTM) is defined in Fig.4 that clearly indicates the out performance of our proposed algorithm. AU-ROC curve when directs close to unity for high TPR and low FPR indicates the algorithm

TABLE III
TECHNICAL COMPARISON OF PROPOSED SYSTEM WITH OTHER STATE-OF-THE-ART WORK

Parameters	Zhou et al. [37]	Tan et al. [38]	Su et al. [39]	Tang et al. [24]	Li et al. [25]	Our Work
Dataset	CICIDS2017	CICIDS2017	NSL-KDD	CICIDS2017	CICIDS2012	CICIDS2017
Algorithm	E(C4.5,RF,ForestPA)	BLSTM	BLSTM	GRU-RNN	RNN,LSTM,CNN	BLSTM
Binary_class	NA	NA	NA	✓	✓	✓
Multi_class	✓	✓	✓	NA	NA	✓
No of Classes	4	8	5	2	2	15
Cuda Enabled	NA	NA	NA	NA	NA	✓
10-fold	✓	NA	NA	NA	NA	✓
Accuracy by Class	✓	✓	✓	NA	NA	✓
Accuracy	97.7%%	NA	99.2%	89%	98%	99.89%
Precision	99.1%	88.86%	NA	99%	NA	99.84%
Recall	NA	91.85%	NA	99%	NA	99.94%
F1-score	99%	90.03%	NA	99%	NA	99.93%
Testing Time/Sample	NA	370(ms)	NA	NA	NA	0.25(ms)
FPR	NA	0.15%	NA	NA	NA	0.15%
others	NA	NA	NA	NA	NA	✓

Others = (TNR, FNR, FDR, FOR, BM, MCC, TS, NPV), NA = Not Available, E(C4.5, RF and Forest PA) = Ensemble(C4.5, RF and Forest PA).

Fig. 5. FNR, FPR, FDR and FOR Rate for Cu-BLSTM, Cu-CNN, Cu-DNN.

efficiency. The AU-ROC curve of Cu-BLSTM demonstrates its efficiency to be used for varied sophisticated intrusions detection.

Computational Complexity

The computational complexity of the proposed algorithm is portrayed in Fig.6. The GPU-Accelerated Bidirectional model of LSTM took 27400 (milliseconds) for almost more than 100 thousand instances in testing phase. Moreover, the testing time for 1 record has additionally been recorded. The computational complexity of Cu-BLSTM for 1 instance is almost 0.25 millisecond which show the speed efficiency of the proposed algorithm. The experiment was also executed on simple CPU to access the performance of deep learning classifiers. The testing time for simple BLSTM was 29800 (milliseconds). Therefore, the proposed algorithm (i.e., Cu-BLSTM) is time efficient. However, it tends to be seen from graph that Cu-BLSTM has a trivial trade-off with Cu-CNN and Cu-DNN in testing time. Consequently, there is a room for improvement to limit the computational complexity of our

Fig. 6. Computational Complexity of Proposed Cu-BLSTM, BLSTM, Cu-CNN, and Cu-DNN.

proposed algorithm.

V. CONCLUSION

With the abstraction power and capability of being innovative, flexible, and adaptive centralized control intelligence of the SD-ITS makes it a promising next generation networking architecture. However, the centralized control unit of the SD-ITS is more likely prone to be compromised being a single point of failure. Moreover, the control plane is having huge potential to be primarily targeted with varied sophisticated evolving multi-vector cyber threats and attacks. This article proposes a DL-Driven highly scalable multi-vector cyber-threat detection framework to protect the control layer of the SD-ITS. Our proposed mechanism outperforms with 99.89% detection accuracy having very low false positives. Further, the proposed technique achieves comparatively low computational complexity. Finally, the authors endorse and tend to investigate ensemble and hybrid DL-driven algorithms for the detection of substantial attacks in different domains of the emerging computing and networking paradigms.

VI. ACKNOWLEDGMENT

This work was supported by the European Commission, under the ASTRID and FutureTPM projects; Grant Agreements no. 786922 and 779391, respectively.

The authors are also thankful to the Deanship of Scientific Research, King Saud University Riyadh Saudi Arabia for funding this research work.

REFERENCES

- [1] A. Akhunzada and M. K. Khan, "Toward secure software defined vehicular networks: Taxonomy, requirements, and open issues," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 110–118, 2017.
- [2] L. Zhao, A. Al-Dubai, A. Y. Zomaya, G. Min, A. Hawbani, and J. Li, "Routing schemes in software-defined vehicular networks: Design open issues and challenges," *IEEE Intell. Transp. Syst. Mag.*, 2020.
- [3] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: A survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.
- [4] Y. Yu, X. Li, X. Leng, L. Song, K. Bu, Y. Chen, J. Yang, L. Zhang, K. Cheng, and X. Xiao, "Fault management in software-defined networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 349–392, 2018.
- [5] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.
- [6] A. Abdou, P. C. Van Oorschot, and T. Wan, "Comparative analysis of control plane security of sdn and conventional networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3542–3559, 2018.
- [7] K. Kalkan, G. Gur, and F. Alagoz, "Defense mechanisms against ddos attacks in sdn environment," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 175–179, 2017.
- [8] R. Swami, M. Dave, and V. Ranga, "Software-defined networking-based ddos defense mechanisms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, p. 28, 2019.
- [9] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325–346, 2016.
- [10] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Communications Surveys & Tutorials*, 2019.
- [11] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [12] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 779–811, 2018.
- [13] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging sdn and nfv security mechanisms for iot systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 812–837, 2018.
- [14] F. Tang, B. Mao, Z. M. Fadlullah, and N. Kato, "On a novel deep-learning-based intelligent partially overlapping channel assignment in sdn-iot," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 80–86, 2018.
- [15] T. Wang, J. Zhou, M. Huang, M. Z. A. Bhuiyan, A. Liu, W. Xu, and M. Xie, "Fog-based storage technology to fight with cyber threat," *Future Generation Computer Systems*, vol. 83, pp. 208–218, 2018.
- [16] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "Iot survey: An sdn and fog computing perspective," *Computer Networks*, vol. 143, pp. 221–246, 2018.
- [17] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in sdn-based industrial internet of things with edge computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.
- [18] J. Wan, C.-F. Lai, H. Song, M. Imran, and D. Jia, "Software-defined industrial internet of things," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.
- [19] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transportation research part C: emerging technologies*, vol. 99, pp. 144–163, 2019.
- [20] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, 2019.
- [21] Q. Li, F. Wang, J. Wang, and W. Li, "Lstm-based sql injection detection method for intelligent transportation system," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4182–4191, 2019.
- [22] Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, and X. Zeng, "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 154 560–154 571, 2019.
- [23] S. Garg, K. Kaur, N. Kumar, and J. J. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in sdn: A social multimedia perspective," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 566–578, 2019.
- [24] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in sdn-based networks: Deep recurrent neural network approach," in *Deep Learning Applications for Cyber Security*. Springer, 2019, pp. 175–195.
- [25] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of ddos attack-based on deep learning in openflow-based sdn," *International Journal of Communication Systems*, vol. 31, no. 5, p. e3497, 2018.
- [26] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep recurrent neural network for intrusion detection in sdn-based networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 202–206.
- [27] C.-H. Huang, T.-H. Lee, L.-h. Chang, J.-R. Lin, and G. Horng, "Adversarial attacks on sdn-based deep learning ids system," in *International Conference on Mobile and Wireless Technology*. Springer, 2018, pp. 181–191.
- [28] R. M. A. Ujjan, Z. Pervaz, and K. Dahal, "Suspicious traffic detection in sdn with collaborative techniques of snort and deep neural networks," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2018, pp. 915–920.
- [29] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21 954–21 961, 2017.
- [30] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (sdn)," *arXiv preprint arXiv:1611.07400*, 2016.
- [31] A. Akhunzada, A. Gani, N. B. Anuar, A. Abdelaziz, M. K. Khan, A. Hayat, and S. U. Khan, "Secure and dependable software defined networks," *Journal of Network and Computer Applications*, vol. 61, pp. 199–221, 2016.
- [32] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K.-K. R. Choo, and J. Iqbal, "A deep cnn ensemble framework for efficient ddos attack detection in software defined networks," *Ieee Access*, vol. 8, pp. 53 972–53 983, 2020.
- [33] A. Akhunzada, E. Ahmed, A. Gani, M. K. Khan, M. Imran, and S. Guizani, "Securing software defined networks: taxonomy, requirements, and open issues," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 36–44, 2015.
- [34] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP*, 2018, pp. 108–116.
- [35] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [36] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2018.
- [37] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, p. 107247, 2020.
- [38] M. Tan, A. Iacovazzi, N.-M. M. Cheung, and Y. Elovici, "A neural attention model for real-time network intrusion detection," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 291–299.
- [39] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset," *IEEE Access*, vol. 8, pp. 29 575–29 585, 2020.