Deep learning approach for cyberattack detection

Conference Paper · April 2018

DOI: 10.1109/INFCOMW.2018.8407032

5 authors, including:

Yiyun Zhou
Kennesaw State University
11 PUBLICATIONS **145** CITATIONS
SEE PROFILE

Meng Han
Georgia State University
64 PUBLICATIONS **632** CITATIONS
SEE PROFILE

Liyuan Liu
Kennesaw State University
16 PUBLICATIONS **141** CITATIONS
SEE PROFILE

Jing He
Kennesaw State University
85 PUBLICATIONS **850** CITATIONS
SEE PROFILE

Deep Learning Approach for Cyberattack Detection

Yiyun Zhou†, Meng Han*,‡, Liyuan Liu†, Jing (Selena) He‡, and Yan Wang‡
‡Graduate College, {yzhou20, lliyuan, ywang63}@students.kennesaw.edu
†College of Computing and Software Engineering, {mhan9, she4}@kennesaw.edu
Kennesaw State University

***Abstract*—With the accelerated growth of internet of things IoT application in recent years, cities have become smarter to optimize resource and improved the quality of life for residents. On the other hand, the IoT face the severe security problem like confidentiality, integrity, privacy, and availability. To prevent the cyberattack irreversible damage, we propose a framework, called DFEL, to detect the internet intrusion in the IoT environment. Through the experimental results, authors present that DFEL not only boosts classifiers' accuracy to predict cyberattack but also significantly reduce the detection time. Furthermore, the paper demonstrates how the DFEL balance the detection performance and speed.**

***Index Terms*—cyberattack, dimension reduction, deep feature embedding learning, real time intrusion detection**

I. Introduction

Cyberattack is a critical problem at the time of Internet of Things(IoT). The research from CSO shows cybercrime damage costs to beat 6 trillion dollars annually by 2021, and the average time for intrusion detection has increased from 57.4 days to 93.2 days in last three years [1]. This paper builds an architecture to prevent the cyberattck in the context of increasing "smart city". Incorporating novel technologies and artificial intelligence, the cities are becoming faster connectivity to optimize resource rapidly in recent years. The "smart city" is defined as using technology to improve city services and provide a high quality of life for residents [2]. Based on the report from Navigant Research, the global market for the smart city is worth 40.1 billion dollars in 2017 and with growth expected to approach a mountainous 97.9 billion dollars by 2026[1].

However, the advanced technologies are under the threat of network intrusion from different aspects. The security and privacy of smart cities' data and information are critical challenges due to the complicated security maturity. In the last ten years, the cyberattack triggered the irreversible damage to our society. In 2013, different groups of hackers compromised 1 billion Yahoo accounts. In 2014, there were 145 million eBay users under cyberattacks. In 2017, the cyber instruction got 143 million consumers personal information from Equifax[2]. The damage from cyberattack never stopped. However, protecting security in the smart city indicates protecting streaming data flow and system from any malicious activity, which needs the costly budget in a long process for public or private sectors [3], [4]. The challenge from cybersecurity not only need the exponential growth in data from IoT but also need an architecture to systematically real-time detect potential threats.

To counter cyberattack in the modern IoT environment, it is critical to strengthening the identification of cyber threats, through executing prompt countermeasures to block the potential risks. Currently, there are some approaches proposed to detect and prevent cyber instructions in the different environment. Ibrahimi *et al.* incorporated the Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to reduce data dimensions to detect intrusions [5], Özçelik *et al.* [6], Ghanem *et al.* [7], and Papamartzivanos *et al.* [8] applied supervised machine learning algorithms, and the neural networks were deployed to solve this problem [9] [10]. The existing method could detect cyberattack with high accuracy on the diverse attack types. However, they barely consider the training time for the massive scale of a dataset. This paper proposes a novel architecture called Deep Feature Embedding Learning (DFEL). This methodology can reduce the data dimensions by taking the edge of deep learning and transfer learning. It also decreases the training time significantly and outperforms other machine learning algorithms concerning the detection accuracy. DFEL can systematically reduce the data dimension by an algorithm 1 by which user can balance the training speed and detection performance of the classifier. Compared with the traditional machine learning methods, the deep learning approach can take advantage of the big data. The fundamental idea of DFEL is to use the mass amount of data to generate high-level features and apply the pre-trained model to boost the detecting speed of traditional machine learning algorithms. Real data experiments show that the DFEL outperforms other up-to-date algorithms when being used on NSL-KDD dataset and UNSW-NB15 dataset. With experiments on the NSL-KDD dataset, the DFEL profoundly improved the gaussian naive bayes classifier's recall level from 80.74% to 98.79% and significantly reduced all classifiers' running time, especially the support vector machine, prediction time from 67.26 seconds to 6.3 seconds. With experiments on the UNSW-NB15 dataset DFEL also discover the critical nonlinear relationships in the data and can help the machine learning algorithms improve performance concerning accuracy and detection time. The section II discusses the current advanced research, section III introduces the methodology of DFEL and section IV evaluates the performance of DFEL on the well known public dataset.

[1]https://www.asme.org/engineering-topics/articles/manufacturing-processing/

[2]https://www.csoonline.com/article/2130877/data-breach/the-biggest-data-breaches-of-the-21st-century.html

978-1-5386-5979-3/18/$31.00 ©2018 IEEE

II. Background and Related Works

There has been abundant literature dealing with cyberattacks detection. Timčenko in [11] compared the ensemble machine learning methods for unbalanced datasets test and indicated the Bagged tree and GentleBoost perform higher accuracy and ROC values than other tree-related values. Ibrahimi *et al.* shows that PCA has high efficiency to discriminate the cyberattacks and standard internet request on KDD-99 and NSL-KDD dataset [5]. Also, the experimental results presented the defect of LDA which has some poor precisely at the level of computations of the covariance matrix. The Özçelik's apply [6] cumulative sum entropy detection to filter the Denial of Service attacks which are an essential problem for the communication systems. The results displayed a high detection accuracy and a low false positive rate. Moreover, the results beat the performances from other detection methods utilizing the entropy of packet header field. In [7], Ghanem implemented Support Vector Machine (SVM) approach, which is a Machine Learning (ML) method and could complement the performance of intrusion detection systems as well as decrease the number of false alarms. Papamartzivanos [8] created a novel methodology for generating new detection rules which could classify both common and rare types of attacks. The performance of the DFEL method outperforms other common ML methods using the KDD-99 and NSL-KDD datasets but still needs to improve on the UNSW-NB15 dataset, which is more complicated and reflects the modern internet traffic.

Deep learning, which has an incredible power to handle image and text data, is evolving extremely fast in recent years. Al-Zewairi [9] applied the Gedeon method to select top 15% essential features and deployed deep learning binomial classifier to detect intrusion. Wang Gang [12] applied the fuzzy clustering technique to generate different training subsets. As a result, training different neural networks on different training subsets could outperform some traditional machine learning algorithms such as decision tree and Naive Bayes. Idhammad [10] utilized unsupervised Correlation-based Feature Selection (CFS) to choose relevant features and built a Feed-forward Neural Network to distinguish the DoS traffic and normal traffic. These research obtained high intrusion detection efficiencies. However, they rarely mentioned the balance between training time and performance. DFEL can significantly reduce the detection time as well as allow the traditional machine learning algorithms take the benefits from big data. Also, DFEL diminishes the data dimension by the algorithm 1 which determines the size of encoding latent variables.

III. Deep Feature Embedding Learning

Deep Learning is a subset of machine learning that consists of supervised and unsupervised learning techniques, and it bases on many layers of artificial neural networks. Each layer contains some neurons with activation functions that are used to produce non-linear outputs. The methodology inspired by the biological neuron structure of the brain, but

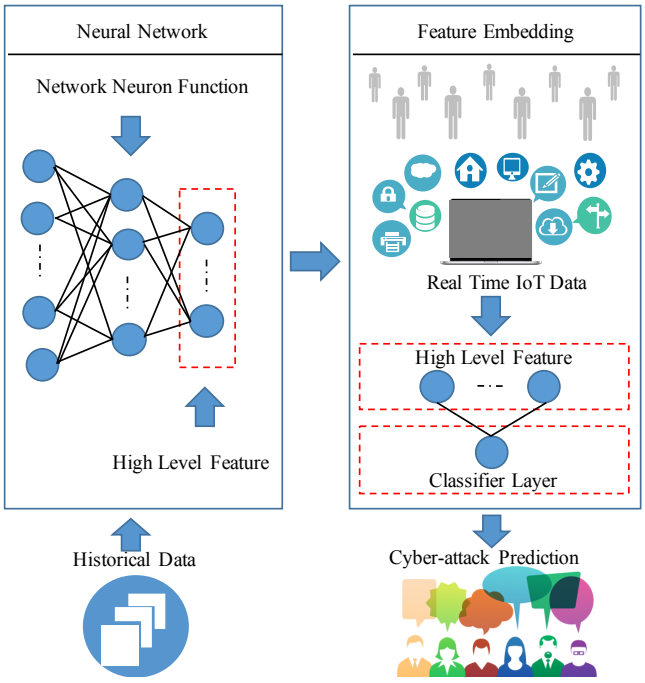

Fig. 1. Architecture for Deep Feature Embedding Learning (DFEL)

it's loosely related to information processing and communication patterns in a natural nervous system. Deep Learning algorithms derive significant abstract representations from the raw data through the use of a hierarchical multi-level learning approach. Features from the higher-level are more abstract and complicated and they based on the less abstract concepts. They are representatives of the features in the lower level of the learning hierarchy [13]. Therefore, the complicated and high-level representatives are valuable as inputs to a supervised predictor [14]. The DFEL method was motivated by these previous researchers as well as by the recent successful implementations of transfer learning in visual categorization [15] and word embedding text analysis [16]. Figure 1 shows the DFEL architecture, which consists of tree stage. In the III-A needs to set the control value (from 0 to 1) to balance the training speed and prediction accuracy. The III-B is training a big dataset on the defined deep learning neural network. In the III-C, pre-trained deep learning network from III-B was used to generate embedding features for small datasets with similar distributions. On the other hand, small-sized datasets can take the advantage from large amount dataset by using the high-level representative features, since this can help the traditional machine learning classifiers to reduce the prediction time and boost the detection accuracy with t.

A. Network Neuron Function

The fundamental objective of the algorithm 1 is to define the number of neurons in each of the hidden layers of the deep neural structure. As a result, the number of neurons in the last hidden layer can be considered as the high-level feature dimension. If the control value ϵ is close to 1, the high-level

feature dimension is more close to the original data dimension. On the other hand, the smaller value of ϵ would generate more abstract representations.

Algorithm 1 Calculate $f(\epsilon) = N$

Require: $D \geq 1$, $0 \leq \epsilon \leq 1$, $N \leftarrow list$
Ensure: $f(\epsilon) = N$
  $i \leftarrow 1$ $\{initial\ i\ equal\ to\ 1\}$
  $l \leftarrow \lceil D/10 \rceil$ $assert\ the\ number\ of\ layers$
  while $N \leq l$ **do**
    $n \leftarrow 0$ $\{initial\ n\ equal\ to\ 0\}$
    $n \leftarrow \lceil (D/i)^{0.5} \rceil$ $\{value\ depend\ on\ which\ of\ layer\}$
    $n \leftarrow n + \lceil \epsilon * D \rceil$ $\{value\ control\ by\ \epsilon\}$
    $N \leftarrow n$ $\{save\ n\ to\ the\ Queue\ N\}$
    $i \leftarrow i + 1$ $\{move\ to\ next\ iteration\}$
  end while

In the algorithm 1, D is the dimension of training dataset, l indicates the number of layers in the Deep Learning Network. In the while loop, number of neurons calculate and save in a list N for III-B. The purpose of algorithm 1 is to determine the number of layers by the dimension of the training dataset. he number of layer's neurons is dcreasing from bottom layers to top layers. The size of neurons is controlled by the user defined variable ϵ, which aims at balancing the detection time and accuracy in III-C. To evaluate how the algorithm can effectly classify network traffic as normal or abnormal network traffic in III-B , authors have set up eleven experiments based on ϵ from 0 to 1 increment by 0.1.

B. Training Deep Neural Network

The deep learning algorithm aims at learning the patterns from the massive amount of dataset. In this study, the deep feed-forward neural networks were trained using the back-propagation algorithm to predict the intrusions. The number of layers and neurons have already been defined by the algorithm 1. The activation function used for each layer is the Leaky Rectifier activation. This non-linear function calculates the sum of all the weighted inputs and transfers the negative values close to zero. The rectifier equation is presented in 1 where the α is 0.001. Compared to the sigmoid activation function or hyperbolic tangent function, the leaky rectifier activation function allows a network to obtain sparse representations as well as relief the gradient vanishing problem.

$$LR(x) = max(\alpha * x, x) \tag{1}$$

The dropout technique regularization technique is applied at each layer to prevent the overfitting problem. Standard backpropagation learning builds up to adapt the training data but does not generalize to unseen data. Random dropout breaks up these adaptations by making the presence of any particular hidden unit unreliable [17]. The output layer uses the sigmoid function to converts the real input value to the monotonical output value that increases from 0 to 1. The output value can be used to classify the intrusion. The function is presented as following 2:

$$S(x) = \frac{e^x}{e^x + 1} \tag{2}$$

The cost function is a binary cross entropy (also called log loss) and measures the intrusion detection probability value between 0 and 1. The function is defined as the followed equation 3, where the y is ground truth and $\hat{y}$ is preditec value, log is nature log.

$$\begin{aligned} arg\ min\ H(y, \hat{y}) = -ylog\hat{y} - (1-y)log(1-\hat{y}) \\ where\ \hat{y} = S(LR^N(x)) \end{aligned} \tag{3}$$

In the backpropagation part, the gradient descent method searches for the optimal global solution that minimizes the prediction error by updating the weights. The partial derivative is calculated for each weight in the network. The learning process can be express as equation 4 where the η is learning rate and t is the number of the epoch.

$$W(t+1) = W(t) - \eta \partial H(t)/\partial W(t) \tag{4}$$

The process of Deep learning network can be presented as algorithm 2. The number layers l and neurons n already calculated in algorithm 1.

Algorithm 2 Calculate $D(x) = xW + b$

Require: N, η,$H(y, \hat{y})$
Ensure: $D(x) = xW + b$
  $Forward\ Propagation$
  $Build\ NN\ with\ l\ layers\ and\ n\ neurons$
  $A(x) = max(\alpha * x, x)$ $\{each\ layer's\ activition\ fuction\}$
  $Drop\ out$ $\{0.1\ drop\ out\ after\ each\ layer\}$
  $S(x) = \frac{e^x}{e^x+1}$ $\{Output\ layer\}$
  $Backpropagation$
  while $validation\ acc\ not\ improve\ in\ last\ 5\ epoch$ **do**
    $W(t+1) = W(t) - \eta \partial H(t)/\partial W(t)$
  end while

After training this model on a large dataset, the output layer was removed and the rest part of the neural network was employed as the pre-trained model for feature embedding. The number of neurons in the last layer of the deep neural network is the encoding variable dimension.

C. Feature Embedding for Detection

Reducing data dimension with the pre-trained deep neural network from III-B can work well if the initial weights were set to close to the optimal solution. Also, this method outperforms the PCA as a tool to reduce the dimensionality of data [18]. The feature embedding for the relatively small size stream IoT data can help classifiers to discriminate the normal traffic and cyberattck. The size of the encoding latent variable dimension is determined by algorithm 1. The smaller size of high-level feature can help classifiers faster detect intrusion, however, may sacrifice the prediction accuracy. Users can balance the detection speed and accuracy by selecting proper

ϵ value. In section IV, the experiment shows the advantage of using DFEL as preprocess for traditional machine learning algorithm.

IV. Data Description and Experiment Evaluation

In this section, authors present the overview of the common cyberattack in IoT environment from the attack features from two public datasets. Then compare the traditional machine learning algorithm detection performance between original dataset and DFEL dataset.

A. Data Collection

- NSL-KDD Dataset: The dataset overcomes the weakness of KDD-99 dataset which was widely used for evaluation of Intrusion Detection Systems [19]. The original KDD-99 dataset contains redundant records in the training dataset, so the classifiers are more likely skewed towards the more frequent history. The cyberattack in the datasets is falling in four categories: Denial of Service Attack (DoS), the user to Root Attack (U2R), remote to Local Attack (R2L). And it's more challenge to detect intrusion using the test datasets since it contains additional more 14 attack types than train data. The essential features extracted from the TCP/IP connection, the traffic features are computed respect to a time window interval.
- UNSW-NB15 Dataset: This dataset created by establishing the synthetic environment at the UNSW cybersecurity lab. The data presented a hybrid of the real modern normal and the contemporary synthesized attack activities of the network traffic [20]. The data had 47 features including new low-footprint attack scenarios and 9 significant families of the cyberattack. UNSW-NB15 is more complicated than NSL-KDD [10]. It is helpful for the research community of network intrusion detection system ($NIDS$) and can be considered as a modern $NIDS$ benchmark data set.

B. Evaluation Metrics

This section mainly aims at evaluating the detection time and accuracy generated by DFEL. Authors also present the detection performance under a set of ϵ value in algorithm 1. In order to nominate the bests strategy for the cyberattack detection, the evaluation metrics are essential. True positive (TP) refer to the correct intrusion detection, and false positive (FP) means to consider the normal traffic as the cyberattack. True negative (TN) corresponds to normal traffic correctly labeled as normal and false negative (FN) refer to fail intrusion disclosure. The experimental results are using the following performance metrics.

- Accuracy: The metric assess the percentage of the internet traffic that is correctly classified. It is a fraction of correction detection divided by the total number of instance in the dataset.
- Recall: This measurement reflects the classifier's ability to detect cyberattack which is vital in our context, also referred as sensitivity.
- Precision: This metric related to the classifier's ability to pass the normal request without condition, also referred as specificity.
- Processing Time Change: The detection time is depend on training time and testing time. The time change (TC) defined as the fraction of classifier detection time without DFEL (T) minus classifier detection time after DFEL (DT) divided by the process time without DFEL.

C. Experiment Result Analysis

Before evaluating the proposed model, all the categorical variable have been transferred to dummy variables. Moreover, min-max normalization was applied on each variable to map the independent variable to the range (0,1). The normalization function is defined as $z = (x - min(x))/(max(x) - min(x))$. The two datasets are randomly splitted using the same rule. 80% of data was used to fit DFEL and get the pre-trained model. The remaining 20% of the data was randomly split into 70%/30% as $training/testing$ data for classifiers. Next, the 20% rest data was transferred to latent attributes using DFEL, and the embedding features were split into 70%/30% for $embedding\ training/embeding\ test$. Finally, the performances from traditional machine learning algorithms were compared on embedding data and original data. In the proposed DFEL model, authors set up eleven experiments based on ϵ from 0 to 1 increment by 0.1 and the highest recall for each classifier was selected.

NSL-KDD Dataset				
Model	Accuracy	Precision	Recall	Time(s)
Gradient Boosting Tree (GBT)	**99.29%**	**99.29%**	**99.29%**	0.41
DFEL GBT (ϵ 0.7)	98.54%	98.54%	98.53%	**0.17**
K-nearest Neighbors (KNN)	98.56%	98.55%	98.56%	1.79
DFEL KNN (ϵ 0.7)	**98.82%**	**98.82%**	**98.82%**	**0.07**
Decision Tree (DT)	90.69%	90.80%	90.61%	0.42
DFEL DT (ϵ 0.7)	**98.77%**	**98.77%**	**98.77%**	**0.25**
Logistic Regression (LG)	95.74%	95.78%	95.71%	0.24
DFEL LG (ϵ 0.7)	**98.85%**	**98.85%**	**98.85%**	**0.20**
Gaussian Naive Bayes (GNB)	81.27%	86.42%	80.74%	0.06
DFEL GNB (ϵ 0.7)	**98.80%**	**98.79%**	**98.79%**	**0.01**
Support Vector Machine (SVM)	94.89%	94.90%	94.87%	67.26
DFEL SVM (ϵ 0.7)	**98.86%**	**98.86%**	**98.87%**	**6.30**

TABLE I
Models Performance Comparison on NSL-KDD Dataset

The performances of the classifiers by using features from original dataset as well as latent features generate by DFEL were compared in the tables I and II.

The machine learning algorithms include gradient-boosted trees, k nearest neighbor, decision tree, logistic regression, gaussian naive bayes and support vector machine. The DFEL approach boosts most classifiers accuracy and significantly saves the cyber detection time. In the experiment on the NSL-KDD dataset, all the classifiers reach the best recall with the DFEL approach by using ϵ of the value 0.7. This metric shows that the most percentages of the intrusion are corrected predicted, which can effectly avoid the potential damage. The accuracy of the gradient boost tree has slightly decreased. However, it reduced the detection time by 57.75% . After the

UNSW-NB15 Dataset				
Model	Accuracy	Precision	Recall	Time(s)
Gradient Boosting Tree (GBT)	**93.13%**	**92.38%**	**92.84%**	0.96
DFEL GBT (ϵ 0.7)	91.22%	90.38%	90.69%	**0.33**
K-nearest Neighbors (KNN)	90.83%	90.06%	90.10%	5.55
DFEL KNN (ϵ 0.7)	**91.90%**	**91.25%**	**91.21%**	**0.16**
Decision Tree (DT)	86.94%	91.28%	82.03%	0.5
DFEL DT (ϵ 0.6)	**92.29%**	**91.24%**	**92.52%**	**0.42**
Logistic Regression (LG)	89.47%	91.02%	86.36%	0.81
DFEL LG (ϵ 1)	**92.35%**	**91.50%**	**92.11%**	**0.42**
Gaussian Naive Bayes (GNB)	50.45%	71.09%	61.18%	0.21
DFEL NB (ϵ 0.7)	**92.52%**	**91.45%**	**92.85%**	**0.02**
Support Vector Machine (SVM)	88.89%	92.25%	84.79%	634.11
DFEL SVM (ϵ 0.6)	**92.32%**	**91.41%**	**92.20%**	**74.81**

TABLE II
MODELS PERFORMANCE COMPARISON ON UNSW-NB15 DATASET

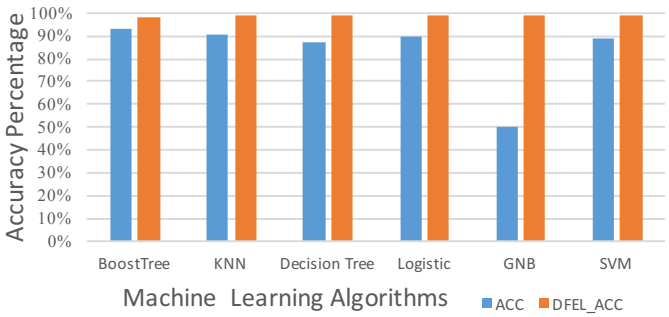

Fig. 2. UNSW-NB15 classifiers' ACC before and after DFEL

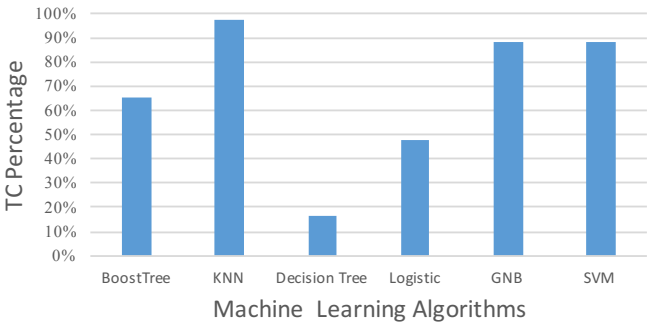

Fig. 3. UNSW-NB15 classifiers' TC after DFEL

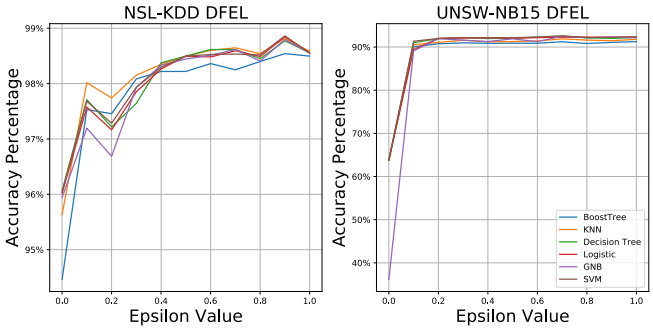

Fig. 4. ϵ effects on accuracy

DFEL process, the gaussian naive bayes classifier's recall level profoundly improved from 80.74% to 98.79%. The support vector machine classifier's detection time reduced from 67.26 seconds to 6.3 seconds. From the accuracy and speed aspect, the DFEL help the traditional machine learning algorithms enhance their performance on NSL-KDD dataset.

The UNSW-NB15 is a more complicated dataset and it is more challenge when being used by classifiers to predict intrusion. As the figure 2 shows, the DFEL discover the critical nonlinear relationships in the data and can help the machine learning algorithms improve performance concerning accuracy and detection time. Support vector machine classifier could increase the recall level from 84.79% to 92.20% and save 559.3 seconds when detecting cyberattack. The gaussian naive bayes classifier's accuracy increased from 50.45% to 92.52% which implicated the DFEL's high-level features represent the original dataset's attributes in a better way. The optimal ϵ value is different than the NSL-KDD dataset, so there is no generalized recommend ϵ for classifiers and data. For UNSW-NB15, the high-level feature generated by DFEL not only well extract the latent insight of the dataset but also reduce the detection time for all classifiers.

The figure 3 presented the detection processing time change for each machine learning algorithm. The decision tree classifier is a fast classier which make prediction based entropy and information gain. So the decision tree classifier doesn't have too much room to speed up. For the time-consuming algorithm like support vector machine and k nearest neighbor, the DFEL significantly decrease the detection time. K nearest neighbor needs to calculate the Euclidean distance to all instances. Core of the support vector machine is a quadratic programming problem which is complexity-wise approximately $\mathcal{O}(n^2_{samples} * n_{features})$. These high computation cost classifiers benefit from the result of dimension reduction performed by DFEL. Furthermore, the proposed framework can also be applied in real-time cyberattack detection.

The figures 4 shows how the ϵ affect the accuracies of classifiers. The ϵ can balance the model performance and cyberattack detection time well. The DFEL with a lower value of ϵ could reduce the dimension of the data more. Therefore, the classifiers can have faster detection time on smaller dimension data. The left apart of figure 4 shows the changing of accuracy with different values of ϵ in the

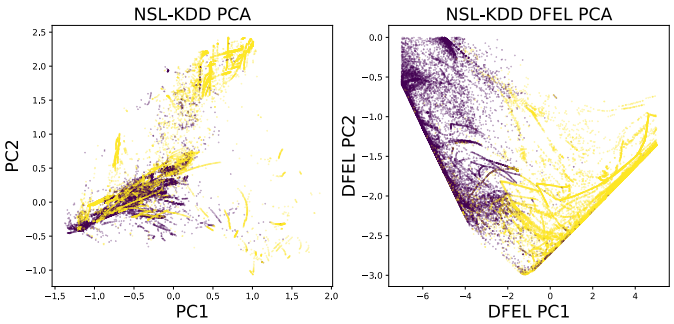

Fig. 5. Comparison PCA visualization before and after DFEL for NSL-KDD

NSL-KDD data. When ϵ changes in the range 0 to 0.4, the accuracies from all the classifiers increase fast. After the 0.4 the accuracy pushes up slowly and close to 1. The right part of figure 4 shows the effect of ϵ on the model performance when using UNSW-NB15 dataset. The accuracy doesn't have significant change after ϵ being above 0.2. To optimize the detection speed and don't sacrifice the prediction accuracy too much, the 0.2 of ϵ value could be the best value for the intrusion detection system.

Figure 5 provides a visualization by using principal component analysis (PCA) dimension reduction. The dark dots are cyberattacks instances and the light dots are normal ones. The PC1 denotes the first principal component, which is the linear combination of attributes that has maximum variance among all linear combinations. The PC2 is the second principal component, which is orthogonal to PC1, measures the remaining variation as much as possible. The left part of figure 5 represents the data attributes in NSL-KDD and the right part reflects the high-level features after DFEL. Before DFEL, the two kind instances mixed together in the left corner and it is difficult for machine learning algorithms to distinguish the differences. Clearly, the DFEL process could well separate the cybetterattack from normal traffic.

In summary, the proposed framework not only reduce the data dimension but also discover the critical nonlinear relationships from the big dataset. The pre-trained model from DEFL could use the ϵ value to optimize the detection time and accuracy of traditional machine learning algorithms. The data visualization with the DEFL features indicates the high-level features are well grouped the cyberattack and normal traffic into two clusters. There are two meaningful observations from our proposed deep feature embedding learning model.

- Adaptability: The DFEL can boost all the classifiers prediction speed and available for real-time detection. Becuase the dimension of the dataset is reduced to a low level, and the less calculation can decrease the algorithm running time. On average it reduces 65.53% time to beat malicious packets. The IDS can balance the algorithm running time and accuracy with the different priority.
- Robustness: The DFEL can help traditional machine learning algorithms to reach the higher accuracy and recall level on various datasets. Becuase the high level features separated intrusion and normal traffic into different clusters are easier for classifiers to converge the optimal solutions.

V. Conclusions and Future Work

This paper proposed a new deep learning approach, DFEL, for real-time cyberattack detection in the IoT environment. The basic idea of this approach is to map the original low level feature d to a high level feature r ($r < d$). In our experiment, the result presents the robustness of high accuracy and significant time-saving. The IDS can trade off the detection time and efficiency with proper settings of ϵ. This approach could also be applied to another context which contains large amounts of data and needs the real-time prediction. In our future work, we will implement this method on real equipment to prevent cyberattacks. On the other hand, we will aim at improving the DFEL abilities in reducing the dimensionalities of the data.

References

[1] M. Nadeau. (2017) State of cybercrime 2017. [Online]. Available: https://www.csoonline.com/article/3211491/security/state-of-cybercrime-2017-security-events-decline-but-not-the-impact.html

[2] C. Cerrudo, "An emerging us (and world) threat: Cities wide open to cyber attacks," 2015.

[3] A. AlDairi *et al.*, "Cyber security attacks on smart cities and associated mobile technologies," *Procedia Computer Science*, vol. 109, pp. 1086–1091, 2017.

[4] M. Han, Z. Duan, and Y. Li, "Privacy issues for transportation cyber physical systems," in *Secure and Trustworthy Transportation Cyber-Physical Systems*. Springer, Singapore, 2017, pp. 67–86.

[5] K. Ibrahimi and M. Ouaddane, "Management of intrusion detection systems based-kdd99: Analysis with lda and pca," in *Wireless Networks and Mobile Communications (WINCOM), 2017 International Conference on*. IEEE, 2017, pp. 1–6.

[6] İ. Özçelik and R. R. Brooks, "Cusum-entropy: An efficient method for ddos attack detection," in *Smart Grid Congress and Fair (ICSG), 2016 4th International Istanbul*. IEEE, 2016, pp. 1–5.

[7] K. Ghanem, F. J. Aparicio-Navarro, K. G. Kyriakopoulos, S. Lambotharan, and J. Chambers, "Support vector machine for network intrusion and cyber-attack detection," 2017.

[8] D. Papamartzivanos, F. G. Mármol, and G. Kambourakis, "Dendron: Genetic trees driven rule induction for network intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 558–574, 2018.

[9] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*. IEEE, 2017, pp. 167–172.

[10] M. Idhammad, K. Afdel, and M. Belouch, "Dos detection method based on artificial neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, pp. 465–471, 2017.

[11] V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," in *Intelligent Computer Communication and Processing (ICCP), 2017 13th IEEE International Conference on*. IEEE, 2017, pp. 13–19.

[12] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert systems with applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[13] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[14] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[15] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[16] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. IEEE, 2009, pp. 1–6.

[20] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Military Communications and Information Systems Conference (MilCIS), 2015*. IEEE, 2015, pp. 1–6.