A GBDT-Paralleled Quadratic Ensemble Learning for Intrusion Detection System

JUN YANG, YIQIANG SHENG, AND JINLIN WANG
National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
Corresponding author: Yiqiang Sheng (shengyq@dsp.ac.cn)

This work was supported by the Chinese Academy of Sciences through the Strategic Priority Research Program under Grant XDC02020400.

ABSTRACT The development of computer and network technology has provided convenience to our daily life, however, attack and intrusion in network emerge endlessly. Intrusion Detection System (IDS) has been developed to confront network attacks. As a result, the research of IDS is one of the most popular fields in recent years. This paper proposes a Gradient Boosting Decision Tree (GBDT)-paralleled quadratic ensemble learning method for intrusion detection system. We use GBDT to deal with the spatial part of traffic data and use Gated Recurrent Unit (GRU) model with special modification for network traffic to deal with temporal data. Then, in order to combine the spatial feature and temporal feature, we fuse GBDT model and GRU model to make a quadratic ensemble model as our final intrusion detection system. The experimental results based on CICIDS2017 dataset show that the advanced spatial-temporal intrusion detection system based on ensemble learning achieves better accuracy, recall, precision and F1 score than the state-of-the-art methods. The accuracies of detecting benign, port scan, Distributed Denial of Service (DDoS), infiltration and web attack traffic are up to 99.9%, 99.9%, 99.9%, 99.9%, and 99.9%, respectively. We also use our method in Information-Centric Networking (ICN) dataset and the results show our method achieves much better performance compared with existing methods.

INDEX TERMS Intrusion detection, deep learning, ensemble learning, model fusion.

I. INTRODUCTION

In recent years, Intrusion Detection System (IDS) is widely used in all respects. Pan et al. [1] developed a hybrid intrusion detection system for power system. Aloqaily et al. [2] built a special intrusion detection system for connected vehicles in smart cities. Ambusaidi et al. [3] created an IDS system to study traffic problem. Hodo et al. [4] tried to use IDS in Internet of Things networks. However, due to the explosive growth of Internet, more and more researchers pay attention on IDS in field of internet security [5]–[8].

Nowadays, the continuous development of computer and network technology has made our life more convenient. Meanwhile, the network-oriented crimes have become more and more frequently, and various types of attacks have emerged in an endless stream. Serious network security events have also been greatly increased. In order to effectively prevent and predict network attacks and security threats, various Intrusion Detection Systems (IDS) [9] have been designed to detect intrusions by analyzing network traffic data. In order to understand the state of the whole network by fusion sensor data from distributed sources, some scholars have proposed the concept of cyberspace situational awareness (CSA) [10], then evaluated the network security in time to identify and predict potential attacks. Nowadays, many types of IDS play an important role in the network security business, which enhances the security of the network and protects users from cyberattacks. Scholars have used various methods to classify attacks, including traditional methods based on hidden Markov model [11], gray Verhulst models [12] and so on.

In recent years, with the prosperity and development of machine learning techniques, more and more researcher try to use machine learning method to deal with IDS problem [13]–[16]. Lots of scholars have used Support Vector Machine (SVM) [17]–[19], Convolutional Neural Network (CNN) [20]–[23], Recurrent Neural Network (RNN) [24], [25], Long Short-Term Memory (LSTM) [26]–[30],
Gated Recurrent Unit (GRU) [34]–[36], ensemble learning method [31]–[33], especially Gradient Boosting Decision Tree (GBDT) [37]–[41] and many other kinds of machine learning methods in IDS. Those methods have efficiently improved the classification accuracy. Among these methods, RNN (including GRU and LSTM) and ensemble learning method (especially GBDT) show strong ability to deal with IDS problem.

RNN is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. Due to this structure, it can show temporal dynamic behaviors. Unlike traditional feedforward neural networks, RNN can utilize their memory (internal state) to create sequences of inputs. RNN can be deemed as multiple copies of the same network, each passing a message to a successor and it is used to deal with natural language modeling in the beginning. However, due to the strong ability to deal with temporal sequence, it is quite popular in cyberattacks field now.

GRU is a successful model to forecast cyberattacks but there is a serious drawback. Compared with traditional ensemble learning models, it is hard to spatial feature because the core idea of RNN is to deal with the temporal relationship in data. In reality, the cyberattacks normally looks like disorganized and fragmented. Sometimes it is nearly impossible to detect the cyberattack by the temporal relationship individually. We also need to rely on the other information to help us identify cyberattacks.

One of the essential characteristics of cyberattack is that the hacker need to use his own computer or a proxy server to start a network attack. No matter how much work to cover up their action, there are some spatial traces left, and we can use IP address, subnet information and network topology information to mining those spatial features. So, it is easy to think that spatial features hide some very useful information of cyberattack. The past researches show that the ensemble learning model has a strong ability to deal with spatial relationship. GBDT is a classical ensemble learning method which combine weak classifier (normally decision tree) to create a strong classifier. GBDT is good at dealing with non-linear relationship, especially fragmented datasets. Because the spatial relationship in the network data is disorganized and fragmented, GBDT has a good potential to detect cyberattacks.

In order to deal with the spatial-temporal relationship in IDS, we create a GBDT-Paralleled Quadratic Ensemble Learning. We use GRU model as our basic model, then we fuse the GBDT and GRU to do quadratic ensemble learning to create a new model. Our quadratic ensemble learning model absorbs the spatial processing capability of GBDT and obtains a better predictive performance than existing models. Compared with the methods in the recent 3 years, our method has achieved state-of-the-art results.

Also, in the majority of studies and researches, the datasets were outdated, including the well-known KDD Cup’99 dataset [42] and ISCXIDS2012 (Intrusion Detection Evaluation Dataset provided by Canadian Institute for Cybersecurity in 2012) [43]. With the change of time, the attack patterns and attack characteristics of traditional cybersecurity datasets differ from today’s new types of cyberattacks, so those outdated datasets are not suitable for today’s cybersecurity research as well.

In 2018, a new IDS dataset called CICIDS2017 (Intrusion Detection Evaluation Dataset provided by Canadian Institute for Cybersecurity in 2017) [44] was proposed by Iman et al from Canadian Institute for Cybersecurity. This dataset includes not only traditional attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), and port scanning, but also some new types of attacks and intrusions, such as the Heartbleed Bug based attack, which is just be found in recent years. Compared to the use of outdated datasets, research on network security and intrusion based on CICIDS2017 can be better applied to today’s network environment.

Recently, many researches focus on Information-Centric Networking (ICN). ICN is one kind of future network architecture which can provide high-quality services such as in the field of video and music for users. However, even though ICN is a very potential network architecture, IDS for ICN is still a new field for research.

The Internet today was built with the narrow waist of IP. In ICN the narrow waist is the contents [45]. In recent years, ICN attracts more and more attention from researchers and tons of ICN projects such as Data-Oriented Network Architecture [46], Network of Information [47], Content-Centric Networking (CCN) [48], Named Data Networking project [49], an extension of the CCN, and the Publish-Subscribe Internet Technologies [50], [51] are developed. All these projects changed the IP narrow waist architecture to make the content as the primary part of communication, and this makes ICN have good scalability. To enhance mobility support, plans such as MobilityFirst [52] was created. The separation of identifier (the name of content) and locator (the network address) makes ICN can have good mobility support. By using the methods such as Distributed Hash Table (DHT), ICN can carry the ever-increasing contents.

Traffic in ICN is very different from the IP network because the primary part of communication is changed from host into content. As it is a new research field, there are few analyses about the attack types in ICN. Research by Compagno et al. [53] points out DDOS attack of interest packets still exists in ICN. Research by Abdallah et al. [54] classifies the ICN attack into four categories, i.e., naming, routing, caching, and other miscellaneous related attacks.

We gathered ICN logs in our lab and make it as a Chinese Academy of Sciences Intrusion Detection Dataset (CAS2018). We use our CAS2018 dataset to test the model’s ability to detect network attacks in ICN.

As we mentioned before, the traditional GRU model has a strong ability to deal with temporal relationship and it gains huge success in the field of detection of cyberattacks. However, there are several drawbacks which can
be improved. In the past research, researchers used one directional GRU to detect cyberattacks. But in reality, the different kinds of cyberattacks may have some hidden relationship. For example, the port scan attacks to target network normally happened after an infiltration attack, so the later information also can help us to discriminate early attacks. In our model, we use Bi-GRU as the substitution of GRU. The result shows that compared with traditional GRU, Bi-GRU has a relatively better ability to detect cyberattack.

Also, GRU cannot deal with the spatial relationship perfectly. There is some method which try to take over the spatial-temporal relationship, but every method has drawback. Hierarchical spatial-temporal features-based intrusion detection system (HAST-IDS) [55] uses the original data flow as input and deems traffic dataflow as a two-dimensional image. This method uses CNN to catch spatial features and uses LSTM to explore the temporal relationship in a network dataset. This structure combined the CNN and RNN into a single model and it is quite popular in Image Caption. The writer uses this structure in cyberattack foresightedly and gets a statistically result. However, it ignores the topology structure in the network so the spatial structure is hard to understand. On the other hand, CNN structure is a black box structure so the explanations is limited and the result is easy to be overfitted. The different characteristics between the image and network flow also reduce the reliability of HAST-IDS.

In order to avoid this problem, we create a new model based on a Bi-GRU and utilize GBDT to enhance GRU. we utilize feature engineering to create a list of features include the information of topology and other spatial features, then do feature selection to choose reliable spatial features.

Our model uses GBDT to enhance the ability to deal with spatial features. the past research had proved that the GBDT is especially good at solving the fragmental spatial information so compared with past models, our model has better spatial processing capacity.

In the end, we combine two different models to make an ensemble learning. GRU is a deep learning method to catch the inner temporal relationship whereas GBDT is an ensemble learning method which is good at deal with fragmented spatial information. These two methods have totally different theory so combining them can absorb their merit in two dimensions (spatial and temporal) and solve the spatial-temporal problem in network problems.

The rest of this paper is organized as follows. Section II describes the related work, including intrusion detection techniques, GRU model, GBDT model, CICIDS2017 dataset and CAS2018 dataset. Section III describes our special feature engineering methods for extracting the spatial-temporal feature of the data, the proposed model in detail and metrics. Section IV contains the evaluation and discussion of our experiment. Section V provides the conclusion of our work and presents future work for researchers.

The contribution of our paper is as followed:

1. We innovatively utilize spatial information to make prediction of cyberattacks. Nowadays, the majority of researchers focus on the temporal relationship in the cyber-attack. Although there are several researchers consider the spatial relationship but they only use the basic spatial feature such as IP address [56] or the potential spatial relationship in original data flow [55] and do not excavate the inner spatial relationship, such as topology relationship. We use a list of spatial features including topology and other spatial information so we can dig the deep spatial characteristic in the network to help us to detect cyberattack.

2. We use an ensemble learning to combine the GBDT model and GRU model as one model. The past research has proved that GRU has a strong ability to catch temporal information in cyberattacks. On the other hand, the design principle of GBDT method is to deal with discrete, fragmental and non-significant information. We use an ensemble learning to absorb these merits so our new method has a strong ability to deal with the spatial-temporal relationship in cyberattacks. Also, the basic idea of GBDT and GRU is totally different so the ensemble method can maximize the merit of two different method and avoid overfitting.

3. Nowadays, ICN becomes a hot research topic. ICN is deemed as one kind of future network architecture, because compared with traditional network, it can provide better service on video or music content for users. However, just as Dainotti mentioned [57], majority institutions do not want to share their traffic datasets because of privacy requirement or other reason. Besides, most famous datasets are based on traditional IP traffic, there are no public traffic datasets based on ICN. In order to test the robustness and generalization of our method, we create an ICN traffic dataset. The result shows that our method has a satisfactory ability to detect cyberattacks.

II. RELATED WORK
A. INTRUSION DETECTION TECHNIQUES

Intrusion detection can be divided into signature-based detection and anomaly-based detection by detection approach [60]. Signature-based detection matches existing rules and recognize existing patterns [61]. The alarm will be activated once a match occurs. However, this method requires a large rule signature database and cannot detect unknown attacks. Anomaly-based detection techniques include traditional probabilistic methods and a new set of machine learning-based methods [62]. In recent years, many scholars have used machine model algorithms to conduct research related to intrusion detection. Wang et al. [17], Thaseen and Kumar [18] and Gu et al. [19] use SVM for intrusion detection, but they simply use the model without taking into account the special effects of the network data itself, resulting in lower detection accuracy. Zhang et al. [20], Potluri et al. [21] and Li et al. [22] use CNN to do the intrusion detection work, but they ignore the temporal features of network data. Wang et al transforms traffic input as a two-dimensional image and uses CNN and LSTM to classify traffic packets [55], but the research
only focuses on the payload of the packet and ignores the space characteristics between the traffic. Hao et al. [24] and Bouzar-Benlabiod et al. [25] use RNN to forecast network attack, but simple RNN cannot achieve a great result. Kim et al. [26], Kim and Kim [28] and Staudemeyer and Omlin [27] use an LSTM-based neural network to process network data, however, LSTM contains too many parameters and has a large computational overhead and they ignore the space characteristics either. Han et al have used Wasserstein Generative Adversarial Network (WGAN) [56] to process data for traffic, nevertheless, models based on Generative Adversarial Network (GAN) requires a very long training time and the model is very hard to train as well. Many scholars used GBDT to deal with network traffic [37]–[41], and get a great result. Hao et al. [34] and Xu et al. [35] used GRU to utilize the temporal relationship of network flow, however, GRU is a version of RNN model which focuses on the inner long-term temporal relationship in dataflow but have a relatively poor ability to catch spatial information. Therefore, we hope to propose a method to comprehensively utilize the spatial-temporal features of network data to complete intrusion detection with good performance.

B. GATED RECURRENT UNIT

Gated recurrent units (GRU) [63] are a special version of recurrent neural network (RNN) [64]. RNN is an extension of a conventional feedforward neural network to handle a variable-length sequence. However, RNN is easy to be influenced by short-term memory. If a sequence is long, RNN is quite hard to transfer useful information form the early sequence to the late sequence. It makes RNN can ignore the important information in a long sequence.

Also, during backpropagation, RNN has serious vanishing gradient problem. Gradient is used to update the weight of our network. If the gradient decay to 0, our network will not update anymore.

LSTM [65] is one of the most popular methods to take over this problem but LSTM has three gates and two memory cells so the training time is quite long when we deal with a huge dataset. Unfortunately, most of IDS datasets are huge so the LSTM model is hard to be trained.

GRU is designed in a manner to have more persistent memory thereby making it easier for RNNs to capture long-term dependencies. Compared with LSTM, GRU has fewer gates and memory cells. Basically, it combines the forget gate and input gate as a single reset gate and merge two memory cells as one. Due to that, the compute of GRU is simpler so it is easy to be trained. The GRU has two gates: reset gate and update gate and one memory. Here is how a GRU use the state and update gate and one memory. Here is how a GRU use the state and update gate and one memory. Here is how a GRU use the state and update gate and one memory. Here is how a GRU use the state and update gate and one memory. Here is how a GRU use the state and update gate and one memory.

\[ r(t) = \sigma \left( W^{(r)} x(t) + U^{(r)} h(t-1) \right) \]  
\[ z(t) = \sigma \left( W^{(z)} x(t) + U^{(z)} h(t-1) \right) \]  
\[ \tilde{h}(t) = \tanh \left( r(t) \cdot U h(t-1) + W x(t) \right) \]  
\[ h(t) = (1 - z(t)) \cdot \tilde{h}(t) + z(t) \cdot h(t-1) \]

As in (1), (2), (3), (4), the responsibility of reset signal \( r(t) \) is to determine how important \( h(t-1) \) is to the computation of new memory \( \tilde{h}(t) \). It calculated by the past hidden state \( h(t-1) \) and the input \( x(t) \). The reset gate has the ability to completely diminish past hidden state if it finds that \( h(t-1) \) is irrelevant to the computation of the new memory.

The update signal \( z(t) \) is responsible for determining how much the past hidden state \( h(t-1) \) will be transferred to new hidden memory \( \tilde{h}(t) \). It calculated by the past hidden state \( h(t-1) \) and the input \( x(t) \) as well. The value of update signal \( z(t) \) is between 0 and 1. If the \( z(t) \) close to 0, the new hidden state will almost equal to new memory \( \tilde{h}(t) \). If the \( z(t) \) close to 1, the new hidden state will almost equal to past hidden state \( h(t-1) \).

The new memory \( \tilde{h}(t) \) is the summaries of the new input \( x(t) \) and the hidden state \( h(t-1) \). The reset signal \( r(t) \) will be used as weight to control how much the hidden state \( h(t-1) \) will be transfer to the new memory. The information of the new memory \( \tilde{h}(t) \) include past information of \( h(t-1) \) as well as input \( x(t) \). GRU use tanh as activation function to squash output to range \([-1, 1]\) as well as zero-center.

The hidden state \( h(t) \) is added by the past hidden state \( h(t-1) \) and the new memory \( \tilde{h}(t) \). The update signal \( z(t) \) control the weight between the past hidden state \( h(t-1) \) and the new memory \( \tilde{h}(t) \).

C. BIDIRECTIONAL GATED RECURRENT UNIT

Bidirectional gated recurrent units (Bi-GRU) are a variant of GRU, which is a combination of a normal GRU and a reverse GRU. At each moment, the input of Bi-GRU will be decided by two different directional GRU. It can catch the spatial influence in data flow by two directions so compared with normal GRU, Bi-GRU has better ability to deal with data with connections between early data and later data.

In the network data, if there are cyberattacks in a data flow, the cyberattacks normally are multiple. The late cyberattacks may imply cyberattacks at early time. Also, in same data flow, the cyberattacks are similar. So, the influence of later data to early data may help us to discriminate cyberattacks. Because of that, we decide to use Bi-GRU in our research.

D. GRADIENT BOOSTING DECISION TREE

Gradient Boosting Decision Tree (GBDT) is a machine learning method for classification and regression problems, which produces a prediction model in the form of an ensemble of basic decision trees, especially Classification And Regression Tree (CART) trees. The idea of GBDT first postulated by Friedman [66], and the functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification. Compared with traditional classifiers, GBDT can produce competitive, robust, interpretable
procedures for both classification and regression, especially
appropriate for mining less than clean data.
Like other boosting methods, GBDT combines decision
trees into a single strong learner in an iterative fashion.
The goal of GBDT is to teach a model \( F \) to predict values of the form \( \hat{y} = F(x) \) by minimizing loss function, such as the
mean squared error (MSE) \( \frac{1}{2} \sum (\hat{y}_i - y_i)^2 \), where \( \hat{y}_i \) is the
estimated value of \( y \) of index \( i \) and \( y_i \) is the actual values of variable \( y \) of index \( i \).

At each stage \( m \), GBDT fit a decision tree \( h_m(x) \) to pseudo-
residuals. Because the dataset used to estimate our model is
limited, only a finite number \( \{F(x_i)\}_1^N \) are involved.

We assume \( F^* (x) = \sum_{m=0}^M f_m (x) \), where the \( f_0 (x) \) is an initial
guess, and \( \{f_m (x)\}_1^M \) defined by the optimization method.
We assume \( J \) is the number of terminal nodes in trees,
which can be adjusted by hand. The coefficients \( J \) controls the
maximum allowed level of interaction in our model. With \( J = 2 \), there are no interaction is allowed. With \( J = 3 \) the model may include effects of the interaction between up
to two variables, and so on. Normally, \( 4 \leq J \leq 8 \) can lead a fairly insensitive to the choice of \( J \) in this range. Let \( J_m \)
be the number of its leaves. The input space into \( J_m \) disjoint regions \( R_{jm} \) , . . . , \( R_{jm} \), and predicts a constant value in every
region. Using the indicator notation, the output of \( h_m(x) \) can
be written as the sum, as in (5),
\[
    h_m(x) = \sum_{j=1}^{J_m} b_{jm} 1_{R_{jm}}(x)
\]
The \( b_{jm} \) is the predicted value in the region \( R_{jm} \).
We can multiply coefficients \( b_{jm} \) by some value \( \gamma_m \) to
minimize the loss function, and the model is updated as in (6), (7).
\[
    F_m(x) = F_{m-1}(x) + \gamma_m h(x)
\]
\[
    \gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h(x_i))
\]

Here the \( \nu \) is the shrinkage to improve the the generalization
ability of model.
Because the loss function \( L(y, F) \) and the base learner \( h(x) \)
is difficult to obtain, given any approximator \( F_{m-1}(x) \), \( \gamma_m \) can be can be viewed as the data-based estimate of \( F^*(x) \)
under the “direction” \( h_m(x) \), a member of the parameterized class of \( h(x) \). It can thus be regarded as a steepest descent step
under that constraint. According to the definition of steepest-
descent, we gain the now equation:
\[
    \rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i))
\]
\[
    F_m(x) = F_{m-1}(x) + \rho_m h_m(x)
\]
We repeat this process until we get final result.

\section{E. Quadratic Ensemble}
Ensemble methods is a kind of learning algorithms that con-
struct multiple classifiers and then make prediction by taking a vote of their predictions. It is a very popular method in
classification. However, in IDS problem, although the spatial
information can be defined, the temporal information is hard
to be identified. In order to combine the temporal information
as well as spatial information, we make quadratic ensemble to
build a new model.
First, we use a traditional ensemble learning method
GBDT to make prediction based on the spatial feature, and
a temporal method bi-GRU to catch temporal information.
then we integrate the GBDT model and the bi-GRU model to
do quadratic ensemble. The detail will be show at figure 2.
Compared with the normal ensemble model or other
traditional classifiers, such as neural net model, quadratic
ensemble is more flexible and can find a balance between
an ensemble of multiple simple classifiers and a single com-
plex classifier so it can achieve a better performance in IDS
problem.

\section{F. CICIDS2017 Dataset}
The CICIDS2017 dataset was used in our study. This dataset
is a network traffic dataset for cybersecurity research, which
is provided by the Canadian Institute for Cybersecurity [35],
containing normal traffic and a range of different types of
attack traffic. In our study, we selected traffic data in the
dataset from July 3rd to July 7th, including benign traffic,
DoS/DDoS attacks, port scans, web attacks, and infiltration
traffic. Each traffic contains pcap traffic file, and over 80
features, including source IP, source port, destination IP,
destination port, number of packets, protocols, and so on.

\begin{table}[h]
\centering
\caption{New labels and old labels of CICIDS 2017.}
\begin{tabular}{|l|l|l|}
\hline
New label & Old label & amount \\
\hline
Benign & Benign & 2359087 \\
\hline
DoS/DDoS Attacks & GoldenEye,DoS & 294506 \\
& Hulk,DoS & \\
& Slowloris,Heartbleed & \\
\hline
Infiltration & Infiltration & 36 \\
\hline
Web Attacks & Brute Force,Web Attack & 2180 \\
& Sql Injection,Web Attack - XSS & \\
\hline
PortScan & PortScan & 158930 \\
\hline
\end{tabular}
\end{table}

Because CICIDS2017 Dataset is very imbalance, the class
imbalance need to be handled before starting experiment
[58], [59]. In CICISD2017 dataset, there are 14 minor class.
We delete 3 uncommon class and merge others into 5 major
class to relieve class imbalance, which show as Table 1. The
3 minor classes we deleted are FTP-Pataor, SSH-Patator as
well as BOT. In real network, FTP transmit user information
by plaintext and has serious security problem so normally it
will be replaced by SFTP or SCP based on SSH. In SSH, due
to security consideration, people usually use RSA key rather
than password login. However, Brute Force is a password-guessing attack so it is not work on RAS key. Even if we use password login, after several wrong password attempt, common tools like fail2ban can ban this IP to connect service to prevent Bruce Force. Because the cost of change IP is quite expensive and Bruce Force is a password-guessing attack which need tons of password attempt so it is easy to be prevented. On the other hand, botnet AREs is a kind of uncommon attack in datasets. In fact, in several famous datasets, such as UNSW-NB15 and CIDDS-001 do not include botnet AREs so we decide to delete this class.

| Traffic Type          | Amount | Rate  |
|-----------------------|--------|-------|
| Benign                | 282341 | 91.43%|
| DoS/DDoS Attacks      | 11541  | 4.26% |
| Port Scan             | 6612   | 1.64% |
| Web Attacks           | 7112   | 1.20% |
| Infiltration          | 4993   | 1.46% |
| Sum                   | 341399 | 100.00%|

However, RAS key can not be Brute Force. Up to bot, it is a kind of uncommon attack and it is not include in majority dataset so we decide to delete them. After class merge, we also resample the dataset. In the end all class in the dataset is over than 1%, the resulting statistics are shown as Table 2.

**G. CAS2018 DATASET**

The CAS2018 dataset was created by our lab.

As Dainotti et al. [57] pointed out, many companies do not want to share their traffic datasets because of privacy and other considerations. Besides, most famous datasets are based on traditional IP traffic, there are no public traffic datasets based on ICN.

**TABLE 2. CICIDS 2017 dataset data distribution.**

In order to test the ability of our model to detect attacks in ICN, we created an ICN traffic dataset. In our lab, there is an experimental ICN environment for some ICN research projects. There are 14 PCs and 42 Sensors linked to our network and they run our special ICN protocol. Every day, sensors collect data and then register those data in our ICN system. As shown in Fig. 1, Computers and sensors are linked to the special ICN Router in our lab, and two infrastructure servers, called Naming Server and Naming Resolution Server (NRS), are available in this network. Naming Server is in charge of naming all the content and devices in this network and Naming Resolution Server is used as a resolution database such as DNS in traditional IP network.

As shown in Fig. 1, in our ICN network, we have an ICN router linking all devices running our ICN protocol. Sensors generate ICN content all the time, including camera data and temperature data. Computers can require ICN content from the ICN network and communicate with each other as well. Naming Server and NRS provide naming and resolution service for other devices. We collect origin data from our ICN network devices and give them a label of type. As in Table 3, there are seven types of traffic in our dataset:

- **Normal Traffic** - Traffic of normal and safe behaviors
- **DDoS to Resolution Server** - DDoS attack traffic to the public NRS. This type of attack can make NRS deny of service and resolution query will timeout.
- **DDoS to Naming Server** - DDoS attack to the public naming server. This type of attack can make NRS deny of service and resolution query will timeout.
- **Flooding Attack** - Attack traffic which can cause network congestion.
- **Jamming** - Attack for jamming the network.
- **Packet mistreatment** - Traffic of abnormal or broken package.
- **Impersonation** - Traffic of using other devices’ private keys without authorization.

The CAS2018 dataset is available at the authors’ Github Homepage (github.com/yangjun1994/CAS2018). Currently, there are 8 files in our dataset, and we may update this dataset in our future experiments. The development of CAS2018 dataset is as shown in Fig. 2.

We run the experiment from 2nd January 2018 to 9th January 2018, and record the logs of our system. All computers run Ubuntu 16.04 as their operating system, and they are connected to the ICN router. Sensors are installed on Raspberry Pi model B with the official Raspbian System. The Raspberry Pi has a wireless adapter, so our sensor data can be transported to the ICN router through the wireless network. The ICN router can make devices only sending packets by
We make feature extraction based on our dataset. First of all, we add the geographic location of both source and destination devices.

III. PROPOSED METHODOLOGY

A. FEATURE ENGINEERING AND DATA PRE-PROCESSING

Because of the connection between network hosts and the relevance of subnets, network traffic data often has certain spatial characteristics. In past research, scholars tend to pay more attention to participate in the optimization of the model, and ignore the study of the network itself. In our research, we analyzed the topology connectivity and host relationship of the network host, adding multi-dimensional feature information of the subnets and spatial location of the traffic connection.

There is also a certain timing correlation between network data. For example, DDoS consists a series of attacks in a short period of time. The past research has proved that the RNN model has a strong ability to identify the inner temporal relationship in cyberattack datasets. Basically, our feature includes the spatial part and temporal part. For the temporal part, we use raw traffic flow as our temporal data $X_{tem}$.

For the spatial data, we create a list of features to help us analysis spatial information.

In the beginning, we analyze the geographical location information of each traffic and create a list of geographic location spatial features. The IP address may hide some very important information of cyberattacks. In a real attack, the resource hacker have is limited so the attack may concentrate on several specific network segment. We parse the IP address of each traffic and create a list of spatial features based on the subnet and host.

More and more researches prove that the topological relationship plays a very essential role in cyberattacks. For example, if device A and device B share the same switcher/router or belong to the same company, we think there are topological relationships between these two computers. The topological relationship shows a potential relationship between different devices. If device A is attacked, device B and the device has a topological relationship with device B has a higher risk to be attacked. The CICIDS 2017 dataset provide a topology relationship [35] and we use its information to make our topology feature. We create a list providing the payload and the name of the receiver, without giving the IP address, and the resolution system are in charge of providing the current address of receiver for packet routing. The files in our dataset are as follows.

“Attack.type” is the definition of the mentioned normal traffic and 6 attack types.

“Computers.na” is the computers’ device name and globally unique identifier (GUID) given by our Naming Server. For example, “Computer1” was given “0cfce03424aa 2ab7ab4999e35c870904534335b” as its GUID.

“Sensors.na” is the sensors’ device name and their GUID given by our Naming Server. In our database, the sensors include cameras, temperature sensors, and humidity sensors. The sensors was installed on many Raspberry Pi, and different sensors on the same Raspberry Pi was given a unique name as a unique content source.

“Devices.key” includes the hash of keys owned by all devices. This file can be used to verify the sender of a packet. If a packet cannot be verified, it may be an attack or a mistreatment packet.

“Publicserver.na” include the address of naming and resolution servers, this file can be used to verify the authenticity of those servers.

“NRS.log” and “NCS.log” are the log file of our resolution and naming servers. Including the timestamp, source address, request content and other essential information.

“Traffic.log” is the log file of all ICN traffic data, including timestamp, source and destination address, packet size, and other traffic features.

The feature extraction of CAS2018 dataset is as following:

Our CAS2018 dataset is based on traffic and log files with logging the timestamp, packet size, packet type et al. We make feature extraction based on our dataset. First of all, based on the information in packages, we make features of packet numbers and bytes per seconds, the duration of the flow, forward inter arrival time and backward inter arrival time et al.

In original traffic records, packet type is only tagged as sig (means signal) or data. Then, we extend the type of signal packets: if a packet is tagged as sig, we will search the server logs to find out which behavior represent this signal packet, and re-tagged it as the real behavior of the signal. (behavior of all legal signal must be one of the following behaviors: register to name server, query from name server, register to resolution server, query from resolution server)

Additionally, in the logs of name server and resolution server, there is a column for user authentication (1 means successful authentication and 0 means fail authentication), so we associated the authentication with traffic information to make new feature.

Finally, there are geographic location of devices in the files named Computers.na and Sensors.na. For each packet, we add the geographic location of both source and destination devices.
of topological spatial feature based on topological relationship between traffic.

Now we have many types of features, including discrete and continuous, numeric and string. The various type of feature increases the difficulty in data processing. In order to deal with these features more robust and effective, we use one-hot strategy and other ways to normalize our features. Although string-like features such as host addresses and protocol ports look like numbers, it is clear that these cannot be directly treated as numbers. The one-hot encoding separates each value of a discrete feature into a single feature so the value of new feature has only 0 and 1, which indicates whether the feature is met respectively. By using one-hot coding, we extend the value of discrete features to the Euclidean space, and a certain value of the discrete feature corresponds to a certain point in the Euclidean space.

After feature engineering processing, we get 80 spatial features. However, there are serious correlations between our features, which may lead to overfitting when we train our model. Also, too many features will increase the dimension of our data and raise the time of model computation.

We use two way to select our spatial feature. First of all, we use a Random forest model to calculate the importance of our feature. We a random choose 50% of all train data and use random forest model to calculate the importance of our feature. we repeat it 100 times and record the importance of our feature every time and rank our feature based on the importance of our feature.

We also use the PCA method to reduce our dimension of spatial feature space. PCA is a classical mathematic method to reduce the dimension of data. The idea of PCA is to reduce the dimension by denoising or eliminating the less important features.

According to these two method, we choose 31 features as our spatial data, we call it $X_{spa}$. Because we use the GRU as our temporal model, so we use a temporal raw dataflow as our temporal data $X_{tem}$.

Through the above feature engineering processing, our data becomes a standardized or encoded data containing comprehensive information of time and space, so that the model construction and training can be carried out.

B. GBDT-PARALLELED QUADRATIC ENSEMBLE LEARNING

GRU is one of the most successful temporal methods and lots of research has proved its ability to deal with temporal information. But as we mentioned before, its theory decides the innate drawback to deal with spatial information. In order to deal with the spatial-temporal relationship in IDS, we create a GBDT-Paralleled Quadratic Ensemble Learning for Intrusion Detection. We select the GBDT model as our spatial classifier and Bi-GRU model as our temporal classifier, the structure of our quadratic ensemble learning is as shown in Fig. 2.

Frist of all, we split our train data $X^{tra}$ as five folds. For every m, we deemed $X^{tra}_{spa}$ and $X^{tra}_{spa}$ as new spatial train dataset. We use this dataset to train our GBDT model and get a GBDT model m, which is our spatial classifier model m. We use this spatial classifier model m to make prediction in $X^{tra}_{spa}$ and get spatial train prediction $Y^{tra}_{spa}$.

For each m, we get a spatial train prediction $Y^{tra}_{spa}$. We combine these 5 spatial train predictions as one final spatial train data $Y^{tra}_{spa}$. To $Y^{tra}_{spa}$, we do total same thing but use the Bi-GRU instead of GBDT model. Similar, we get a final temporal train data $Y^{tra}_{tem}$. We combine $Y^{tra}_{spa}$ and $Y^{tra}_{tem}$ and get our final train data $Y^{tra}$.

Up to our test data $X^{tes}$, just as what we do before, we split it as five folds. For each fold m, we use the spatial classifier m and temporal classifier m we get before to make prediction. Please note that we do not train the again but use the trained model to make prediction. In the end, we can get a spatial test prediction $Y^{tes}_{spa}$ and a temporal test prediction $Y^{tes}_{spa}$.

We calculate the average of $Y^{tes}_{spa}$ and $Y^{tes}_{spa}$ separately and get $Y^{tes}_{spa}$ and $Y^{tes}_{spa}$. We combine the spatial dataset and temporal dataset then we get $Y^{tes}$. Now we have a train dataset
and a test dataset, we use them to train a final meta classifier (here we use LightGBM method) and use this method to make our final prediction in $Y_{tes}$. In this way, we get a GBDT-Paralleled Quadratic Ensemble Learning method which takes into account both the spatial-temporal feature of the network data. The Algorithm show the concrete step and the table of notation (Table 4) show the explanation of our symbols used in Algorithm.

Algorithm 1 GBDT-Paralleled Quadratic Ensemble Learning

Input: raw data $X_{raw}$;
Output: Predicted class;
1: do feature engineering and get spatial feature and temporal feature form CICIDS 2017;
2: do feature selection and get original dataset $X$, $X = \{X_{spa}, X_{tem}\}$;
3: split dataset $X$ as train data $X_{tra}$ and test data as $X_{tes}$;
4: split $X_{tra}$ as five folds $\{X_{m}\}_{m=1}^5$, $X_{tes}$ as five folds $\{X_{m}\}_{m=1}^5$;
5: for each fold $X_{tra}$, split $X_{tra}$ as $X_{spa}$ and $X_{tra}_{tem}$ we get $X_{tra}_{spa} = \{X_{spa}\}_{m=1}^5$ and $X_{tra}_{tem} = \{X_{tem}\}_{m=1}^5$;
6: for $m = 1$ to 5 do
7: deem $X_{tra}_{spa} - X_{tra}_{spa}_{m}$ as new spatial train dataset $X_{spa}_{m}$;
8: use $X_{tra}_{spa}$ to train GBDT model, get GBDT model $m$;
9: use GBDT model $m$ to make prediction in $X_{tra}_{spa}_{m}$, get spatial train prediction $Y_{tra}_{spa}_{m}$;
10: end for
11: let $Y_{tra}_{spa} = \{Y_{tra}_{spa}_{m}\}_{m=1}^5$ as final spatial train data;
12: for $m = 1$ to 5 do
13: deem $X_{tra}_{tem} - X_{tra}_{spa}_{m}$ as new temporal train dataset $X_{tem}_{m}$;
14: use $X_{tem}_{m}$ to train Bi-GRU model $m$;
15: use Bi-GRU model $m$ to make prediction in $X_{tem}_{m}$, get temporal train prediction $Y_{tem}_{m}$;
16: end for
17: let $Y_{tra}_{tem} = \{Y_{tra}_{tem}_{m}\}_{m=1}^5$ as final temporal train data;
18: let $Y_{tra} = \{Y_{tra}_{tem}, Y_{tra}_{spa}\}$;
19: for each fold $X_{tes}$, split $X_{tes}$ as $X_{tes}_{spa}$ and $X_{tes}_{tem}$, we get $X_{tes}_{spa} = \{X_{tes}_{spa}\}_{m=1}^5$ and $X_{tes}_{tem} = \{X_{tes}_{tem}\}_{m=1}^5$;
20: for $m = 1$ to 5 do
21: use GBDT model $m$ to make prediction in $X_{tes}_{spa}_{m}$, get spatial test prediction $Y_{tes}_{spa}_{m}$;
22: use Bi-GRU model $m$ to make prediction in $X_{tes}_{tem}_{m}$, get temporal test prediction $Y_{tes}_{tem}_{m}$;
23: end for
24: calculate the average of $\{Y_{tes}_{spa}_{m}\}_{m=1}^5$, get final spatial test data $Y_{tes}_{spa}$;
25: calculate the average of $\{Y_{tes}_{tem}_{m}\}_{m=1}^5$, get final temporal test data $Y_{tes}_{tem}$;
26: combine final temporal test data and final spatial test data, we get $Y_{tes} = \{Y_{tes}_{spa}, Y_{tes}_{tem}\}$;
27: use $Y_{tra}$ to train meta classifier (We use GBDT here);
28: use meta classifier to make final prediction in $Y_{tes}$;

Table 5 and table 6 show the hyperparameter search ranges for our classifier and the optimum hyperparameter values in the final model.

Table 5. Hyperparameter search ranges for GBDT model and the optimum hyperparameter values in the final model.

| Hyperparameter               | Search range       | Optimized value |
|------------------------------|--------------------|-----------------|
| Number of leaves             | [32, 128]          | 64              |
| Learning rate                | [0.001, 0.1]       | 0.01            |
| Number of estimators         | [100,200000]       | 10000           |
| colsample_bytree             | [0.5, 1]           | 0.7             |
| min_split_gain               | [0.0, 0.1]         | 0               |
| min_child_weight             | [0.0, 0.1]         | 0               |
| min_numb_rounds              | [10,1000]          | 100             |

Table 6. Hyperparameter search ranges for GRU model and the optimum hyperparameter values in the final model.

| Hyperparameter               | Search range       | Optimized value |
|------------------------------|--------------------|-----------------|
| Learning rate                | [0.001, 0.1]       | 0.01            |
| L1 regularization penalty    | [0.001, 0.1]       | 0.001           |
| L2 regularization penalty    | [0.001, 0.1]       | 0.002           |
| units                        | [20,1000]          | 100             |
| batch_size                   | [10,100]           | 25              |

C. METRICS

True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are used for evaluation the model performances. (see Table 7).

The explanation is as follows:

True Positive (TP): An attack in actual is classified as an attack in prediction.
TABLE 7. Confusion matrix.

| Actual Class | Predicted Class | 
|--------------|----------------|
| Attack       | TP             |
| Benign       | FN             |
| Benign       | FP             |
| Attack       | TN             |

True Negative (TN): Benign in actual is classified as benign in prediction.

False Positive (FP): Benign in actual is classified as an attack in prediction.

False Negative (FN): An attack in actual is classified as benign in prediction.

Based on the above confusion matrix, we can calculate Accuracy, Precision, Recall and F1 Score to evaluate the performance of our model.

Accuracy: Accuracy, as in (10), is one of the most common performance metrics used in the evaluation of models. It is defined as a ratio of correctly predicted traffic to all traffic.

\[
\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (10)
\]

Precision: Precision, as in (11), is the ratio of correctly predicted positive attack traffic to the total predicted positive attack traffic. High precision means the low false positive rate. This a metric to judge the ability to distinguish all positive attacks.

\[
\text{Precision} = \frac{TP}{(TP + FP)} \quad (11)
\]

Recall: Recall, as in (12), is the ratio of correctly predicted positive attack traffic to all traffic in actual class. This a metric to judge the correctness to distinguish positive attacks.

\[
\text{Recall} = \frac{TP}{(TP + FN)} \quad (12)
\]

F1 score: F1 score, as in (13), can be thought of as a weighted average of the model precision and recall, with a maximum of 1 and a minimum of 0. Therefore, it takes both false positives and false negatives into account.

\[
F1 \text{Score} = 2 \ast (\text{Recall} \ast \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (13)
\]

IV. EVALUATION

We run model on a workstation with Intel® Core™ i9-7900X CPU @ 3.30GHz, 128 GB RAM, 1TB SSD, 4TB HDD, Nvidia® Titan™ XP and Nvidia® Tesla™ P-100 GPU. The detail will show in Table 8. The runtime of our model is 85min.

We used SVM model as classical model, LSTM, Bi-LSTM, GRU, Bi-GRU model as traditional temporal methods, LightGBM [67] and eXtreme Gradient Boosting (XGBoost) [68] as the best traditional ensemble model with great classification result. and 7 state-of-the-art methods to compare the performance with our proposed method.

The 7 state-of-the-art methods are as follows:

LSTM-Attention was proposed by Lin et al. [69] in Jul 2019. They used an LSTM model to build a deep neural network model and added an attention mechanism to enhance the performance of the model.

HAST-IDS was proposed by Wang et al. [55]. They used the CNN model and the LSTM model to build a deep neural network model, and they got a very good result.

Serpil et al proposed a Shallow Neural Network (SNN) and Deep Neural Network (DNN) model with autoencoder to detect cyberattack [70] in Jun 2019. Their DNN model with autoencoder got the best performance in their research work.

Razan et al proposed a feature dimensionality reduction approach for machine learning [59] in Feb 2019. They used a random forest (RF) model with the Principle Component Analysis (PCA) and they developed Uniform Distribution Based Balancing (UDBB) method.

Attention Flow-WGAN [56] was an existing fusion model of intrusion detection system using WGAN and attention mechanism to process flow data, proposed by Han et al in June 2019, which is the newest research with great performance and state-of-the-art result.

Reinforcement Learning-based Intrusion Detection System (RL-IDS) was a big data-driven IDS method proposed by Otoum et al. [71], and it got extremely good result on big sensed data for intrusion detection.

Deep belief and Decision Tree-based Hybrid Intrusion Detection System (D2H-IDS) [2] was a combination of Deep Belief Network and Decision Tree, it was developed by Moayad Aloqaily et al. It was build as an intrusion detection system for connected vehicles in smart cities AND the performance on NSL-KDD dataset was nearly perfect.

The performance of our method and the above methods for detecting different kind of traffic are as shown in Table 9, Table 10, Table 11, Table 12, Table 13.

| Parameter name | Parameter content |
|----------------|------------------|
| CPU            | Intel® Core™ i9-7900X @ 3.30GHz |
| Memory         | 128 GB           |
| Storage        | 1TB SSD and 4TB HDD |
| GPU #1         | Nvidia® Titan™ XP |
| GPU #1 Memory  | 12 GB            |
| GPU #2         | Nvidia® Tesla™ P-100 |
| GPU #2 Memory  | 16 GB            |
| Operating System | Ubuntu 18.04 LTS |
| CUDA Version   | CUDA 10.0       |
| cuDNN Version  | cuDNN 7.4       |
| Tensorflow Version | Tensorflow 1.13.1 |

Table 9 shows the experimental performance on benign traffic. It shows the comprehensive ability to detect network attacks. Compared with other models, almost all of our model’s performance is better than others. Our model achieved the highest accuracy score, recall score, F1 Score and the second highest precision score in all models. Comprehensively, the performance of our model is best above in all models. The classical model SVM got lowest score in all score.
In the CICIDS2017 dataset, the PortScan attack will occur after infiltration attack. The attack will execute the Portscan attack by the victim’s computer. According to the result showed by Table 10, almost all model except DNN with autoencoder model achieves a satisfactory result (all judgment score is over 0.95). The performance of our model is best among the selected model and reaches 0.999 in all score. LightGBM, XGBoost and RL-IDS also get nearly perfect results. It implies that the ensemble model has a great ability to deal with PortScan data. On the other hand, the performance of models with spatial structure shows the bidirectional structure may have slightly merit to catch spatial-temporal relationship.

A DDoS attack generates the huge network data flow to overwhelm the targeted system and make the online service unavailable from multiple sources.

Up to table 11, the performance of our model to classify DDoS attack is extremely high. LSTM-Attention model and Attention Flow-WGAN also got really good performance. Pca-rf with UDBB model achieved good precision (0.978) but the Recall rate was extremely low (0.078). On the other hand, the model with temporal structure got stability result than ensemble models. The ensemble models (LightGBM and XGBoost) achieved good precision score (0.931 and 0.952) but got relatively low result in Recall score (0.697 and 0.745). It shows the ensemble learning model have the ability to detect the DDoS attack but may misclassify the normal data flow or other types of attacks to DDoS attack. The reason may be that there is inner spatial relationships in DDoS attack. As we know, a DDoS attack always happened frequently in short time, and the destination of IP is the same. In this case, if we can use the temporal and spatial structure.
information to make prediction, the DDoS attack is not hard to be determined. However, a simple ensemble model is hard to deal with this information so the result is not good as the model with spatial structure.

Another interesting thing to note is that the model with Bidirectional structure have better performance than the model without these structures. the average of scores in Bi-LSTM and Bi-GRU was slightly higher than LSTM and GRU. It may imply that the bi-direction may has a better ability to catch the inner spatial feature in DDoS influence.

The infiltration of the network from inside is normally exploiting a vulnerable software such as Microsoft Word. Infiltration attacks will leave leak on target’s computer after successful exploitation and start various attacks on the target’s network such as IP sweep, full port scan and service enumerations.

Table 12 shows the performance to determine cyberattack on infiltration traffic. The performance of all models in accuracy is very good (higher than 0.98). However, the majority of spatial models had relatively low score in precision, recall as well as F1 Score. The reason could be that the feature of infiltration attack will not work until the successful exploitation so there is not strong spatial regulation. Also, the feature of the infiltration attack is scattered but may have some general characteristics. Ensemble learning models are good at dealing with these characteristics so it is no doubt that LightGBM and XGBoost models have much better results than spatial models.

Up to our model, the performance of our model in accuracy and precision is better than others although Attention Flow-WGAN achieved the best result in recall score. Because of the prediction of Attention Flow-WGAN model based on the original data flow, which may exist some general characteristic including the hidden information of infiltration attack.

Web attack is a common attack type. In CICIDS 2017 dataset, SQL injection is used to make web attack. SQL injection creates a string of SQL commands then use these commands to force the database to reply the information. If the developers don’t test code properly to find the possibility of script injection, the Cross-Site Scripting will happen and Brute Force over HTTP can try a list of passwords to find out the administrators’ passwords.

Table 13 shows the performance of methods on web attack traffic. Our model achieved the highest accuracy, precision, recall, F1 score in all models. It shows that our model can detect web attacks effectively.

On the other hand, bi-spatial models and ensemble models both reached relatively good scores in accuracy and precision but the ensemble learning models got much better results in recall and F1 score. Because web attacks normally happened very frequently in a short time, which like the DDoS attack, the spatial model may misclassify these two attacks and lead a poor result of recall score and F1 Score.

Another interesting thing is that the accuracy of the pca-rf with UDBB model, DNN with autoencoder and Attention Flow-WGAN also reached very high level. However, these models got very low score in both precision, recall and F1 score. It means these model focus detection of web attack and misclassify lots of normal traffic as web attacks. So, the comprehensive classification ability is quite low even it got very high score in accuracy.

On the other hand, Ensemble models show a strong ability on accuracy and precision and acceptable ability on recall and F1 score. It shows that ensemble models will not misclassify normal data flow as attack. The models with spatial structure also got acceptable result in accuracy score although that the Recall scores was relatively lower than ensemble models. On the other hand, the performance of bi-models in Precision score is obviously better than models without bi-structure.

Overall, spatial model and ensemble models have their own advantage to deal with different attacks. Our model fused these two models’ merit together and got the best result among these models in CICIDS 2017 dataset.

However, the traditional dataset is a little obsolete to face future challenge. Nowadays, in order to take over this problem, we created an ICN dataset and used it to make prediction.

### TABLE 14. Performance of proposed method for CAS 2018 on normal traffic and comparison with other methods.

| Method       | Accuracy | Precision | Recall | F1 Score |
|--------------|----------|-----------|--------|----------|
| SVM          | 0.926    | 0.720     | 0.679  | 0.699    |
| LSTM         | 0.945    | 0.812     | 0.741  | 0.775    |
| Bi-LSTM      | 0.948    | 0.826     | 0.747  | 0.785    |
| GRU          | 0.945    | 0.811     | 0.743  | 0.776    |
| Bi-GRU       | 0.948    | 0.828     | 0.743  | 0.783    |
| LightGBM     | 0.952    | 0.833     | 0.778  | 0.804    |
| XGBoost      | 0.954    | 0.939     | 0.792  | 0.814    |
| Proposed Method | 0.979    | 0.928     | 0.902  | 0.915    |

### TABLE 15. Performance of proposed method for CAS 2018 on DDoS to resolution server and comparison with other methods.

| Method       | Accuracy | Precision | Recall | F1 Score |
|--------------|----------|-----------|--------|----------|
| SVM          | 0.973    | 0.717     | 0.484  | 0.578    |
| LSTM         | 0.986    | 0.833     | 0.793  | 0.813    |
| Bi-LSTM      | 0.988    | 0.872     | 0.799  | 0.833    |
| GRU          | 0.986    | 0.823     | 0.791  | 0.807    |
| Bi-GRU       | 0.981    | 0.847     | 0.611  | 0.710    |
| LightGBM     | 0.995    | 0.952     | 0.935  | 0.943    |
| XGBoost      | 0.981    | 0.847     | 0.611  | 0.710    |
| Proposed Method | 0.995    | 0.910     | 0.904  | 0.907    |

As shown in Table 14, Table 15, Table 16, Table 17, Table 18, Table 19, Table 20, we have 7 different classes in our ICN dataset, including Normal Traffic, DDoS to Resolution Server, DDoS to Naming Server, Flooding Attack, Jamming, Packet mistreatment, Impersonation. Because lots of methods do not suit for ICN dataset, we only used 7 traditional method and our methods to make comparison. The result shows a similar conclusion that we got in the CICIDS 2017 dataset.

Compared with other models, our method reached a significantly better result in the CAS2018 ICN dataset. It implies that our method is robust and stable in different datasets and can be used in future network architecture like ICN.
TABLE 16. Performance of proposed method for CAS 2018 on DDoS to naming server and comparison with other methods.

| Method    | Accuracy | Precision | Recall | F1 Score |
|-----------|----------|-----------|--------|----------|
| SVM       | 0.967    | 0.746     | 0.504  | 0.601    |
| LSTM      | 0.979    | 0.809     | 0.758  | 0.783    |
| Bi-LSTM   | 0.950    | 0.813     | 0.753  | 0.790    |
| GRU       | 0.980    | 0.818     | 0.763  | 0.790    |
| Bi-GRU    | 0.982    | 0.858     | 0.762  | 0.785    |
| LightGBM  | 0.977    | 0.868     | 0.631  | 0.731    |
| XGBoost   | 0.975    | 0.831     | 0.625  | 0.713    |
| Proposed Method | 0.992 | 0.919     | 0.917  | 0.918    |

TABLE 17. Performance of proposed method for CAS 2018 on flooding attack and comparison with other methods.

| Method    | Accuracy | Precision | Recall | F1 Score |
|-----------|----------|-----------|--------|----------|
| SVM       | 0.995    | 0.704     | 0.770  | 0.736    |
| LSTM      | 0.997    | 0.774     | 0.859  | 0.815    |
| Bi-LSTM   | 0.997    | 0.779     | 0.875  | 0.824    |
| GRU       | 0.997    | 0.782     | 0.871  | 0.824    |
| Bi-GRU    | 0.997    | 0.785     | 0.873  | 0.827    |
| LightGBM  | 0.998    | 0.904     | 0.894  | 0.899    |
| XGBoost   | 0.998    | 0.859     | 0.892  | 0.875    |
| Proposed Method | 0.999 | 0.914     | 0.929  | 0.921    |

TABLE 18. Performance of proposed method for CAS 2018 on jamming and comparison with other methods.

| Method    | Accuracy | Precision | Recall | F1 Score |
|-----------|----------|-----------|--------|----------|
| SVM       | 0.999    | 0.895     | 0.853  | 0.875    |
| LSTM      | 0.999    | 0.918     | 0.910  | 0.914    |
| Bi-LSTM   | 0.999    | 0.935     | 0.912  | 0.923    |
| GRU       | 0.999    | 0.908     | 0.908  | 0.912    |
| Bi-GRU    | 0.999    | 0.924     | 0.912  | 0.918    |
| LightGBM  | 0.999    | 0.918     | 0.910  | 0.902    |
| XGBoost   | 0.999    | 0.902     | 0.902  | 0.902    |
| Proposed Method | 0.999 | 0.974     | 0.963  | 0.968    |

TABLE 19. Performance of proposed method for CAS 2018 on packet mistreatment and comparison with other methods.

| Method    | Accuracy | Precision | Recall | F1 Score |
|-----------|----------|-----------|--------|----------|
| SVM       | 0.975    | 0.140     | 0.408  | 0.209    |
| LSTM      | 0.994    | 0.636     | 0.560  | 0.596    |
| Bi-LSTM   | 0.994    | 0.668     | 0.603  | 0.634    |
| GRU       | 0.994    | 0.644     | 0.561  | 0.600    |
| Bi-GRU    | 0.995    | 0.675     | 0.613  | 0.642    |
| LightGBM  | 0.996    | 0.764     | 0.729  | 0.746    |
| XGBoost   | 0.996    | 0.773     | 0.717  | 0.744    |
| Proposed Method | 0.998 | 0.874     | 0.863  | 0.869    |

TABLE 20. Performance of proposed method for CAS 2018 on impersonation and comparison with other methods.

| Method    | Accuracy | Precision | Recall | F1 Score |
|-----------|----------|-----------|--------|----------|
| SVM       | 0.990    | 0.743     | 0.736  | 0.740    |
| LSTM      | 0.994    | 0.834     | 0.819  | 0.827    |
| Bi-LSTM   | 0.994    | 0.839     | 0.836  | 0.838    |
| GRU       | 0.993    | 0.825     | 0.807  | 0.817    |
| Bi-GRU    | 0.994    | 0.837     | 0.830  | 0.834    |
| LightGBM  | 0.994    | 0.836     | 0.835  | 0.835    |
| XGBoost   | 0.994    | 0.836     | 0.833  | 0.835    |
| Proposed Method | 0.998 | 0.933     | 0.928  | 0.931    |

V. DISCUSSION

As shown in Fig. 4 and Fig. 5, our model achieves best result among these models. The reason of this result may be the advantage of catching long-term spatial-temporal relationship in the network.

On the other hand, the traditional SVM method has relatively poor performance in our dataset compared with other method. The temporal methods (LSTM, Bi-LSTM, GRU and Bi-GRU) get a very good result on port scan. It is interesting to note that the performance of bi-structure is obvious better than normal structure on web attack. The reason may be that webattack normally happened very frequently and there are several different type attacks happened after a successful webattack. Because of that, the later attack can offer very useful information to detect webattack.

Attention Flow–WGAN is a WGAN model with attention mechanism, this method shows a relatively good ability to detect all type of attack. Actually, the performance of this model and HAST-IDS is only next to our model.

DNN with autoencoder shows a good ability to catch spatial information so it gets a very high result on portscan and DDoS attack. But its structure focus on the spatial features and do not consider an inner temporal relation between attack so it hard to detect webattack as well as infiltration.
HAST-IDS combines CNN model and LSTM model so this method has good ability to deal both spatial information and temporal information. Although this model is not the best one on any particular kind of attack but the performance is very stable and acceptable.

LSTM-Attention is a LSTM model with attention mechanism, it has a very strong ability to deal with temporal information. However, it does not include a Bi-structure so its performance on webattack was very poor.

Pca-rf with UDBB achieves a good result on DDoS attack but as a whole it is not a successful method in CICIDS 2017 dataset.

D2H-IDS has a good performance, however the ability to detect web attack is relatively low, the reason maybe is that this structure is designed as an IDS for connected vehicles so it may ignore some essential feature for web attack.

RL-IDS has very good performance on DDoS and protoscan attack. However, because it does not include a temporal method, so this model is not the best structure to deal with time-related attack, such as infiltration and webattack.

The performance of LightGBM and XGBoost was similar. Compared with other model, their ability to detect DDoS attack is not good. The reason is that these methods will misclassify the normal data flow or other types of attacks to DDoS attack.

Overall, it is clear that the performance of the method with spatial structure is better than others. It may imply that spatial information can help us to detect cyberattacks. Also, it is interesting to notice that the performance of the spatial models with two direction are slightly better than models without two direction.

One of the most essential characteristics in network dataset is huge and multifarious, which lead an increasing danger to overfitting. Our model used an ensemble learning method to predict cyberattack, which has a strong ability to deal with the overfitting problem. Also, according to feature selection, we limited the number of features and reduced the correlations between features, which decrease the proximity of overfitting as well as increase the reliability of our result.

GRU model is a kind of neural net model and the theory is far different from the GBDT (which the core theory is the ensemble of weak classifier). Compared with an ensemble of two similar models, our ensemble model can absorb the merit of two model and achieved a better result than any single model.

VI. CONCLUSION AND FUTURE WORK
In order to utilize exist spatial-temporal relationships in network data, we created a GBDT-Paralleled Quadratic Ensemble Learning for Intrusion Detection. Past researches showed that the GRU model has a memory structure and has the ability to capture long-term dependencies. However, it is hard to catch the spatial feature. Unlike the traditional way, we used a GBDT model to enhance the ability to catch spatial-temporal relationship. Due to the merit to deal with disorganized and fragmented data, GBDT has a good potential to catch spatial features in cyberattacks. According to ensemble the GBDT model and the traditional GRU model, we took over the problem of spatial-temporal relationship in network flow and we got a great result on the CICIDS2017 dataset and CAS2018 dataset. Compared with other models, our model can identified the inter influence of spatial relationship and had best ability to predict cyberattacks.

However, the traffic of ICN has some unique characteristics. In traditional IP network architecture, IP address represents both identifier and locator. On the other hand, in ICN architecture the identifier and the locator are not bonded. For example, in traditional IP network architecture, if a user moves from A to B, the identifier and the locator will change together but in ICN architecture, only the locator will change because the identifier only relates to the user. This feature lets ICN architecture has a bright prospect on application scenario with mobility requirement, such as 5G and Internet of Vehicles scenario. In our CAS dataset, we gathered the ICN traffic in our experiment network under these considerations, but in reality, production network is much complex than the experiment network in our lab. So in the future, we will evaluate and optimize our model in a more complex ICN network environment.

ACKNOWLEDGMENT
The authors would like to thank Yiran Hao, Luchao Han, Kankan Zhao, and Hongyu Zhou, and thank the editors and reviewers for insightful comments.

REFERENCES
[1] S. Pan, T. Morris, and U. Adhiakari, “Developing a hybrid intrusion detection system using data mining for power systems,” IEEE Trans. Smart Grid, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.
[2] M. Aloqaily, S. Otonn, J. I. Adhawhi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” Ad Hoc Netw., vol. 90, Jul. 2019, Art. no. 101842.
[3] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” IEEE Trans. Comput., vol. 65, no. 10, pp. 2986–2998, Oct. 2016.
[4] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, “Threat analysis of IoT networks using artificial neural network intrusion detection system,” in Proc. Int. Symp. Netw. Comput. Commun. (ISNCC), May 2016, pp. 1–6.
[5] A. Javed, Q. Niyazi, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in Proc. 5th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol. (Formerly BIOMETICS), 2016, pp. 21–26.
[6] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goo, L. T. Heberlein, C.-L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, and D. Mansur, “DIDS (distributed intrusion detection system)-motivation, architecture, and an early prototype,” Comput. Secur. Lab., Division Comput. Sci. Univ. California, Davis Davis, CA, USA, Tech. Rep., 1991. [Online]. Available: http://dl.lib.mrt.ac.lk/bitstream/handle/123/12232/DIDS_Distributed_Intrusion_Detection_Sys.pdf?sequence=1&isAllowed=yes
[7] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, “Fuzziness based semi-supervised learning approach for intrusion detection system,” Inf. Sci., vol. 378, pp. 484–497, Feb. 2017.
[8] A. A. Abruomman and M. B. I. Reaz, “A novel SVM-kNN-PSO ensemble method for intrusion detection system,” Appl. Soft Comput., vol. 38, pp. 360–372, Jan. 2016.
[9] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016.
Y.-B. Leau and S. Manickam, “A novel adaptive grey verhulst model for network security situation prediction,” Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 1, pp. 90–95, 2016.

B. A. Tama and K.-H. Rhee, “An in-depth experimental study of anomaly detection using gradient boosted machine,” Neural Comput., vol. 31, no. 4, pp. 955–965, Apr. 2019.

I. H. Sarker, Y. B. Abusharkh, F. Alsolami, and A. I. Khan, “IntrudTTree: A machine learning based cyber security intrusion detection model,” Symmetry, vol. 12, p. 754, May 2020.

A. Verma and V. Ranga, “Machine learning based intrusion detection systems for IoT applications,” Wireless Pers. Commun., vol. 111, no. 4, pp. 2287–2310, Apr. 2020.

A. Verma and V. Ranga, “On evaluation of network intrusion detection systems: Statistical analysis of CIDS-001 dataset using machine learning techniques,” Pertanika J. Sci. Technol., vol. 26, no. 3, pp. 1307–1332, 2018. [Online]. Available: https://dx.doi.org/10.11111/9827918.17416

Y. Hao, Y. Sheng, and J. Wang, “Variant gated recurrent units with chi-square feature selection and multi class SVM,” J. King Saud Univ.-Comput. Inf. Sci., vol. 29, no. 4, pp. 462–472, Oct. 2017.

J. Gu, L. Wang, H. Wang, and S. Wang, “An novel approach to intrusion detection using SVM ensemble with feature augmentation,” Comput. Secur., vol. 66, pp. 53–62, Sep. 2019.

S. Chasen and C. A. Kumar, “Intrusion detection model using fuzzy inference system for anomaly detection using chi-square feature selection and multi class SVM,” J. Eng. Comput. Math., vol. 7, no. 1, pp. 1–15, Apr. 2017.

M. Zhang, B. Xu, S. Bai, S. Lu, and Z. Lin, “A deep learning method to detect Web attacks using a specially designed CNN,” in Proc. Int. Conf. Neural Inf. Process. Cham, Switzerland: Springer, 2017, pp. 828–836.

S. Potluri, S. Ahmed, and C. Diedrich, “Convolutional neural networks for multi-class intrusion detection system,” in Proc. Int. Conf. Mining Intel. Knowl. Explo. Cham, Switzerland: Springer, 2018, pp. 225–238.

Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, “Intrusion detection using convolutional neural networks for representation learning,” in Proc. Int. Conf. Neural Inf. Process. Cham, Switzerland: Springer, 2017, pp. 858–866.

J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, “CNN-based network intrusion detection against denial-of-service attacks,” Electronics, vol. 9, no. 6, p. 916, Jun. 2020.

Y. Hao, Y. Sheng, J. Wang, and C. Li, “Network security event prediction based on recurrent neural network,” J. Netw. New Media, vol. 6, no. 5, pp. 54–58, 2017.

L. Bouzar-Benlabiod, L. Meziani, S. Rubin, K. Belaïdi, and N. E. Haddar, “Variational encoder-decoder recurrent neural network (V-EDRNN) for anomaly prediction in a host environment,” in Proc. IEEE 20th Int. Conf. Inf. Reuse Integrate Data Sci. (IRI), Jul. 2019, pp. 75–82.

Y. Gu, Y. Gu, W. Liu, and J. Peng, “XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud,” in Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), Jan. 2018, pp. 251–256.

A. Bansal and S. Kaur, “Extreme gradient boosting based tuning for classification in intrusion detection systems,” in Proc. Int. Conf. Adv. Comput. Data Sci. Singapore: Springer, 2018, pp. 372–380.

M. K. S. S. B. Kri, “Grey-based network intrusion detection using clustering and gradient boosting,” in Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (INCCNT), Jul. 2018, pp. 1–7.

Z. Wang, E. Jiang, Y. Gu, W. Liu, and J. Peng, “XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud,” in Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), Jan. 2018, pp. 251–256.

A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” Comput. Secur., vol. 31, no. 3, pp. 357–374, May 2012.

I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116.

Information Centric Networking (ICN). CISCO, San Jose, CA, USA, 2015. [Online]. Available: https://www.researchgate.net/profile/ David_Oran/publication/267534332_Information-Centric_Networking_Is_It_Ready_for_Prime_Time_Will_It_Ever_Be/links/ 54eb9f180cfd20828518e4038/Information-Centric-Networking-Is-It- Ready-for-Prime-Time-Will-It-Ever-Be.pdf

T. Koponen, M. Chawla, B.-G. Chun, A. Ermlincksky, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” ACM SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 181–192, Oct. 2007.

C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, “Secure naming techniques,” in Proc. Int. Conf. Platform Technol. Service (PlatCon), Feb. 2019, pp. 1–6.

V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT), 2009, pp. 1–12.

L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. C. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, E. Yeh, “Named data networking (NDN) project,” Xerox Palo Alto Res. Center-PARC, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010, p. 158.

J. Yang et al.: GBDT-Paralleled Quadratic Ensemble Learning for IDS

IEEE Access, vol. 8, pp. 75–84, Apr. 2020.
