IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 November 2023, accepted 3 December 2023, date of publication 12 December 2023,
date of current version 18 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3341911

RESEARCH ARTICLE

Malicious Data Classification in Packet Data Network Through Hybrid Meta Deep Learning

SAKIB UDDIN TAPU[1], SAMIRA AFRIN ALAM SHOPNIL[1], RABEYA BOSRI TAMANNA[1], M. ALI AKBER DEWAN[2], (Member, IEEE), AND MD. GOLAM RABIUL ALAM[1], (Member, IEEE)

[1]Department of Computer Science and Engineering, BRAC University, Dhaka 1212, Bangladesh
[2]School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB T9S 3A3, Canada

Corresponding authors: Sakib Uddin Tapu (sakib.uddin.tapu@g.bracu.ac.bd) and Samira Afrin Alam Shopnil (samira.afrin.alam.shopnil@g.bracu.ac.bd)

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC), Canada; and in part by BRAC University, Bangladesh.

ABSTRACT Advancements in wireless network technology have provided a powerful tool to boost productivity and serve as a vital communication method that overcomes the limitations of wired networks. However, because of using wireless networks, security is an increasing concern in the community. At the time of our study, people rely on machine learning techniques to create a trustworthy networking system. However, it hinders the development of a reliable network as the number of publicly available malicious data is insufficient to train a model correctly. In real life, people are not very keen to share this data as they are sensitive. In order to deal with this issue, we primarily aim to develop a solution that provides a reliable intrusion detection system despite being trained with a small amount of data. This paper proposes a novel idea of hybrid meta deep learning in detecting malicious packet data. We use a combination of Siamese and Prototypical networks where the Siamese network is used for binary classification and the Prototypical network for multi-class classification. Both approaches are based on meta learning techniques, requiring a minimal amount of data for most attack classes. Utilizing these meta learning characteristics, we could train our model with just 3000 data samples and achieve more than 90% accuracy for both meta learning tactics. Our study aims to provide a secure and trustworthy network domain that enhances communication between end users.

INDEX TERMS CSE-CIC-IDS2017, CSE-CIC-IDS2018, few-shot learning, hybrid meta learning, intrusion detection, malicious data classification, meta learning, multi-class classification, prototypical network, Siamese network.

I. INTRODUCTION

Communication has always been a fundamental part of life for human interaction. People are always looking for an efficient and effective communication medium. Following that legacy, we entered into wireless communication from wired. Nowadays, the emphasis on wireless technology is progressing at a rapid pace. While enjoying the benefits of wireless communication, we also have face malicious attacks. As cyber attacks evolve, attackers exploit our system with many vulnerabilities, which may hamper our system on a large scale. Currently, most conventional machine learning solutions for intrusion detection rely heavily on a more significant number of samples and unfortunately publicly accessible network data is scarce. Moreover, a systematic approach to anomaly detection with a limited volume of attack samples is subsequently essential. Thus, attackers can easily bypass the system's security with advanced behavior to accomplish their objective. Furthermore, with the substantial development of new attacking strategies, the number of attacks is increasing and their patterns are also changing. Therefore, researchers are currently working to develop

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

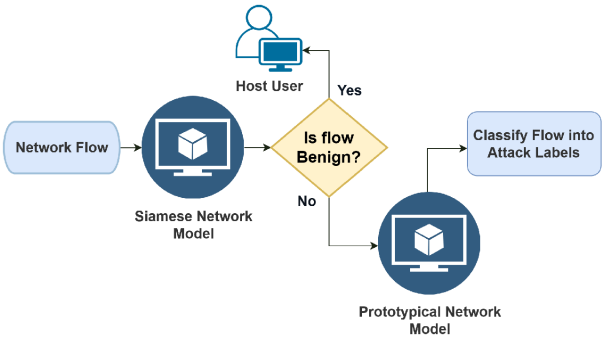


FIGURE 1. High level diagram of our hybrid meta learning approach.

a method that can recognize existing assaults from their patterns and foretell other attacks that they have never encountered. Developments to date have highlighted two dominant challenges and the importance of developing an anomaly detection method that addresses the stated problems. The first need is a robust model that can be trained with a small number of instances and still predict any threat and the other is a model that should be able to protect the system from the emergence of any unknown abnormal behavior.

The focus of this study is to develop a trusted platform between users that allows them to communicate with the highest level of information security more appropriately than current state-of-the-art networks. In this work, we aim to address an existing requirement: introducing models that identify the malicious attack in the packet data flow and classify them into multiple classes with minimal data as the number of publicly available datasets for malicious data is very limited. Moreover, the state-of-the-art methods can mainly distinguish up to 8 different classes with many samples. Thus we took the initiative to solve the challenges of detecting more malicious attacks with smaller datasets. The block diagram of our proposed model is illustrated in Fig. 1. It represents the simplest form of our proposed architecture. At first, the network flows are sent to the Siamese network. Here, binary classification is being done and from the outcomes of the Siamese network, we can determine whether a network flow is malicious or not. The network traffic is then forwarded to the user if it is benign. Additionally, if the network data is suspected to be malicious, it is passed to the Prototypical network for further analysis and possible classification into malicious classes.

The contributions of this work are summarized as follows:

- We proposed a Hybrid Meta Deep Learning based approach for malicious data classification, which can classify 15 distinct labels with greater than 90% accuracy and F-1 score while being trained with only 3000 data samples.
- In the hybrid meta deep learning technique, the Prototypical network and the Siamese network are combined in order to construct a hybrid meta learning based model that will ensure secure user communication. Meta learning enables improved accuracy with much fewer training data.
- At the time of our research, we were unable to locate any papers using hybrid meta learning in the field of network security. In order to address the scarcity of malicious network traffic in a network environment, we took the initiative to introduce a hybrid meta learning method for intrusion detection.
- We present several analyses to demonstrate the efficacy of our method for identifying 15 different classes using CSE-CIC-IDS2017 and CSE-CIC-IDS2018 datasets.
- We present a comparative analysis of our research with some of the current state-of-the-art intrusion detection models to show the effectiveness of Hybrid Meta Deep Learning.

II. RELATED WORK

Due to the trivial amount of literature related to meta learning in the cyber security domain, it is beyond the scope of this paper to provide an extensive background of affiliated work. However, there will be overwhelming evidence that a trusted platform is needed for PDN/network data. Furthermore, there will be ample proof that a trusted platform can be achieved by hybrid meta learning methods with limited samples to send packet data. The following literature review will briefly describe the fact.

The core process of deep learning includes feature extraction and by using that a neural network can be trained to learn any function and make predictions. In [8] Kim et al. developed a deep learning based intrusion detection model, especially for identifying denial of service or DoS attacks. In their paper, they used CNN for binary and multiclass classification and evaluated their model, compiling all the DoS attacks from CSE-CIC-IDS 2018 and KDD CUP 1999. In [11] Kim et al. illustrated deep-learning approaches and constructed a convolutional neural network (CNN) for intrusion detection. They presented results for each subset of CSE-CIC-IDS 2018 dataset, containing up to three different types of malicious attacks along with benign or non-malicious behavior. Their results show that the model performs well only when a large amount of data is present. In [22], Lu and Ding proposed a semi-supervised deep learning model to address the issues of expensive labeled data and the challenge of labeling samples in supervised learning. Here, they combined supervised learning and unsupervised learning using a ladder network. However, their algorithm could not identify any unknown or encoded network data. A Deep Neural Network based intrusion detection system was presented by Ishaque et al. in [23]. They made use of the UNSW NB15 dataset for binary and multiclass classification, where nine different attacks were present. However, they did not give the complete outcome of each malicious attack and only used accuracy and precision as evaluation metrics for their model. In [16], the authors proposed a deep learning based IDS/IPS method for DoS attacks. Their developed technique aims to determine whether a packet is

malicious and stop the attack from causing damage. The CICDDOS2019 dataset is used to classify the traffic into malicious and benign categories.

Meta Learning is a technique that allows models to learn how to learn. This approach enables models to understand the problem better with much fewer data. In [7], Bie et al. demonstrated the importance of a meta learning based method for network intrusion detection. This research presented MultiBoosting multi-classifiers to significantly enhance the detection performance of traditional machine learning intrusion detection methods. However, their model still requires a large amount of data to gain the detection rate they achieved. In [9], Hindy et al. introduced a Siamese Network model employed as the One-Shot learning architecture to classify five types of cyber attacks. Moreover, the network's performance on classifying a new cyber-attack class without re-training is assessed where the amount of newly labeled attack classes is very small. In the study [10], Wang et al. talked about a Siamese network combining attention-based mechanisms for few-shot learning. Here, they learned embedding functions using CNN networks and used conventional attention kernel functions to compare the similarities of two feature vectors. This demonstrates the effectiveness of attention mechanisms in a Siamese network. In [20], Wang et al. suggested a method called ID-FSCIL to classify newer attacks appropriately with smaller data sets by modifying the original attack detection approach. They added meta learning to the currently used machine learning techniques to find novel attacks. To distinguish malicious network data from regular network traffic, in [21], Wu et al. developed a Convolutional One-Dimensional Siamese neural network or COD-SNN. They evaluated their model using the UNSW-NB15 dataset containing nine distinct malicious attack classes and normal network data for binary classification. In [24], Xu and Wang suggested a metric-based first-order meta-learning framework that allow the training of intrusion detection models across various tasks to optimize the model's generalization capacity. The trained model can rapidly adjust to new attacks with a few shots of samples. However, they did not provide a comprehensive breakdown of the results of their model for detecting specific attack categories. Using a few-shot learning approach, in [12] Park et al. developed a Siamese Convolutional Neural Network (Siamese-CNN), demonstrating excellent outcomes with only a minimal amount of training data. However, they are only classifying up to 8 different attack types. To increase the semantic discriminability between prototypes, in [19], Mo et al. suggested a concise contrastive learning approach that employed a metric loss in the Siamese style. They carried out comprehensive evaluations on numerous benchmarks, and the results depict the effectiveness of visual presentation for image classification. The authors in [18] proposed a few shot learning based network intrusion detection model that comprised of a feature extraction network called F-Net and a comparison network called C-Net. They performed binary classification on two datasets, namely ISCX2012 and CICIDS2017. However, specific attack related countermeasures cannot be taken as the classification indicates whether a network is benign or malicious. A few-shot learning-based Siamese capsule network was created by Wang et al. in [13] to address the lack of training data for anomalous network traffic and improve the detection of suspicious threats. In addition, the Siamese network was effectively incorporated with an unsupervised sub-type sampling technique to enhance the detection of network intrusion attempts. Their experimental findings were demonstrated with a relatively small number of samples. However, their model classifies only 8 different attack types from the CIC-IDS-2017 dataset. The authors in [28] suggested a model to detect unseen malware variants using few shot learning techniques. Here, they combined multiple datasets to have a variety of malware data and benign data. They performed binary classification to classify the data into benign or malicious class. Miao et al. in [25] presented a few-shot traffic multi-classification model called SPN. Here they used the architecture of Siamese Network to develop a Prototypical Network to distinguish 8 distinct malicious classes using CSC-CIC-IDS2017 dataset.

Hybrid meta learning is the combination of two or more meta learning techniques into a single architecture. By incorporating a Prototypical network module into a Siamese network, the authors in [14] Wang and Zhai discussed a few-shot learning architecture that uses Euclidean distance to learn high quality prototype representations of each class. By performing image classification, they claimed that the suggested architecture could assist the model in generalizing to new classes not included in the training set, despite only several samples of each class. This paper signifies the importance of combining the Prototypical and Siamese networks as a hybrid tool in effectively identifying different classes with fewer data samples. A Siamese-prototype network with prototype self-calibration and inter-calibration for few-shot remote sensing image classification was discussed in [15] by Cheng et al. To get suitable prototypes, they calibrated the ones produced from support features using the supervision knowledge from support labels. Then, they considered how confidence scores interact between the support and query samples to calibrate the prototype further. This paper further proves how a Prototypical and Siamese network combination performs exceptionally well even with few data samples. In [26] Yang et al. proposed a multi-tiered hybrid intrusion detection system that combines a signature-based IDS with an anomaly-based IDS to identify cyber-attacks on vehicular networks. A signature-based IDS was utilized in the proposed architecture to identify and categorize known attacks, while an anomaly-based IDS was used to distinguish between normal and attack network flow. After balancing they generalized the attacks and evaluated their model for 6 different attack types. In the study [27], the authors introduced a meta learning ensemble model to do binary classification. They used a combination of stacked

ensemble and MLP meta-estimators to detect malicious attacks. They considered all types of malicious data as attack and their model was used to differentiate between benign and malicious attack.

In [3], Zhijun et al. primarily concentrated on low-rate DoS attacks and their mechanism and try to determine the LDoS attack generation concept. By doing rigorous analysis, they figured out that feature detection approaches are more suited to identifying LDoS assaults due to the high positive rate. This backs up our approach of using a feature extractor like CNN to extract the features from data. In order to categorize the obstacles with the existing security models and to generate new directions for security framework developments using effective ML or DL methods, Jayalaxmi et al. provided helpful information in [4] for industry and academia. They built a strong foundation for a prediction model and offered a blueprint for a mapping technique for analyzing the level of risk. Additionally, an integrated multilevel hybrid architecture was suggested for future development by combining signature and anomaly detection with risk factor mapping. In [2] Shaukat et al. discussed a concise overview of machine learning techniques and how they have been or may be used to identify and categorize cyberattacks and spam filtering on mobile and smartphone devices as well as computer networks. They mentioned that there is a need for techniques that not only detect a higher number of attack types but do it with fewer samples. In the survey [5], Khamaiseh et al. thoroughly analyzed the adversarial attack techniques and how they function. They also presented a rigorous analysis of the attack technologies. In [6], Wazid et al. were concerned about the 5G network system and other allied fields to thoroughly analyze the domain's future developments. Some of the difficulties they cited include the need for protocols to defend against multiple attacks simultaneously, the need for low computation power, low communication costs, and small storage sizes without compromising system security, the need for protocols to operate in complex environments while maintaining high scalability, the need for protocols to support a variety of devices and the mechanisms connected to them, and others. In [17], Verma et al. analyzed intrusion detection systems that mainly focused on confronting attacks. They explained how the different types of attacks work. Moreover, they stated that deep learning has drastically improved upon the shortcomings of Machine learning.

III. METHODOLOGY

The following section describes the process we used to reach the results shown. This section presents our overall methodology, how to prepare the data to meet the specifications for our architecture, and an in-depth look into how our architecture works.

A. OVERVIEW OF THE PROPOSED METHOD

This subsection demonstrates the overview of the proposed method, which is also depicted in Fig. 2.

The publicly available datasets CSE-CIC-IDS2017 and CSE-CIC-IDS2018 were first collected from the University of New Brunswick website. Following that, the datasets are preprocessed to ensure that there are no null or infinite values, and the classes are labeled. Afterward, the data is transformed into images of dimension 13×6 to extract features using 2D CNN layers. After splitting the data into training and test sets, data is passed to the Siamese network. The training image dataset is converted into the augmented dataset D_{Train}, where each data point contains two images. If both images are benign or malicious, then the data point is labeled as 1, and if the images are different from each other, it is labeled as 0. This dataset is then used to train the Siamese network to optimize its parameters for a similarity function. Next, the augmented test dataset D_{Test} is created where each data point contains one random benign image from the training set and another random image from the testing set. This creates a dataset where each data contains a benign image from the train set and another image the model has not seen before. Labeling of data is done the same way as before. The model is then evaluated on D_{Test}.

The data points classified into malicious data are then used to train and evaluate the Prototypical network. The training malicious data images are first converted into 14-way k-shot support and query sets. Both of these sets are then passed to the model for training to optimize the feature extractor. Afterward, we create another 14-way k-shot support set using images from the training data and a 14-way 1-shot query set using images from the testing data. They are then used to evaluate the performance of the trained Prototypical network.

B. MODEL ARCHITECTURE

The following subsection provides a comprehensive study of the methodology of the proposed Hybrid Meta Deep Learning techniques.

1) META LEARNING

Meta learning is one of the most promising and exhilarating research domains in the AI field. Meta learning breaks the traditional model training method with huge training datasets. It introduces the idea of training a model on various related tasks with fewer data samples, and it can use this learning for related new future tasks. This allows us to deploy machine learning techniques in domains with minimal data. Few shot learning is one such technique of meta learning.

2) FEW SHOT LEARNING

Few shot learning is a learning method where the training dataset contains limited data. It is also known as n-way k-shot learning, where k denotes the number of data points of n classes. Few shot learning models have been developed to accomplish work with a constrained amount of training samples to address the identification of unknown classes.

3) SUPPORT SET

The term ''support set'' is used in meta learning. Let us take a dataset called D and randomly select some data samples

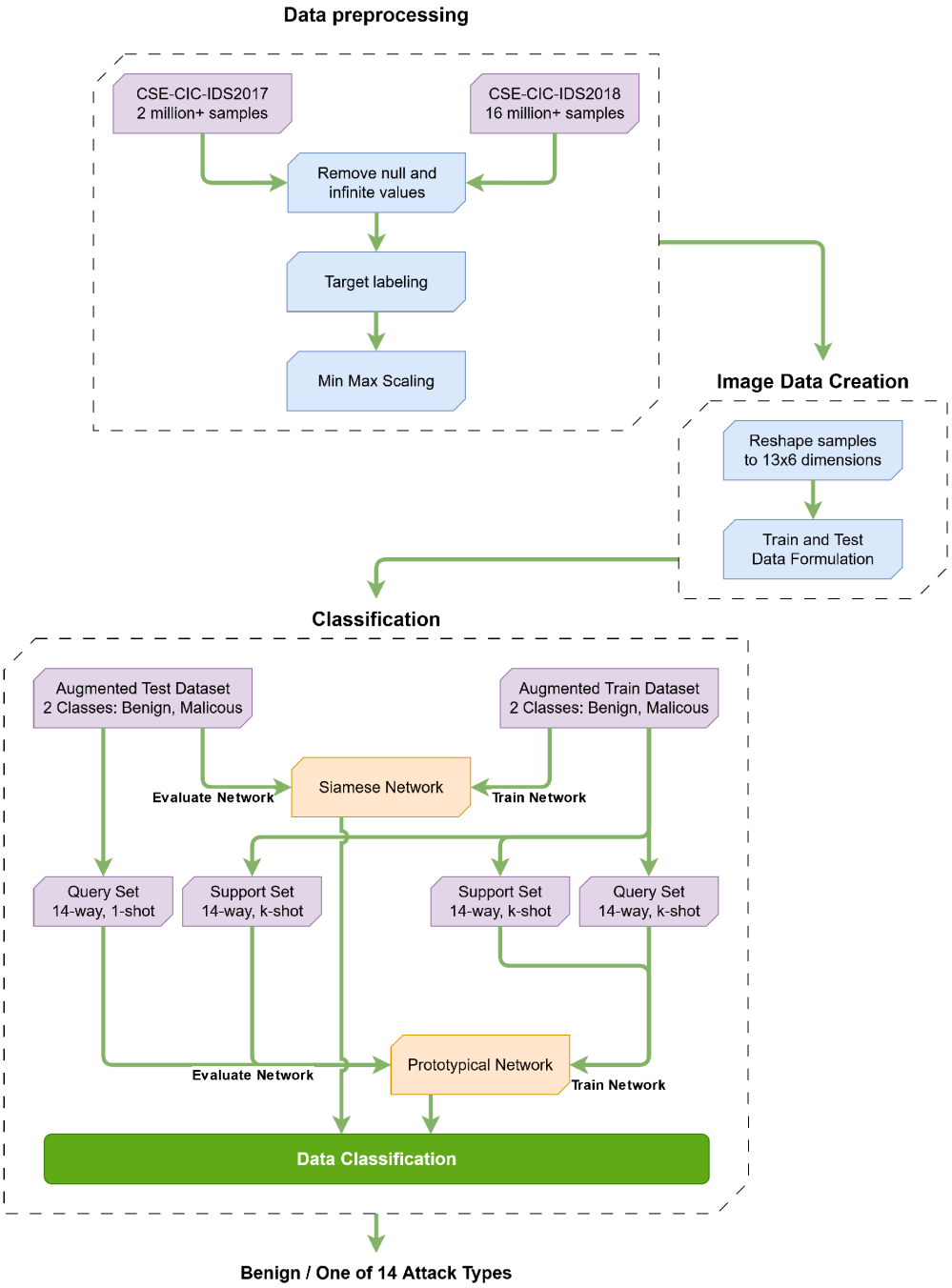


FIGURE 2. Top level overview of the proposed framework.

from each class without replacing them to create a different dataset. That dataset will be regarded as a support set. An N-way K-shot support set will contain N different classes of data, and each of the N classes will have K number of data points. For example, a 5-way 2-shot support set will have five data classes, and each class will have 2 data points for a total of $5 \times 2 = 10$ data points. Data from this set provides the network with support information that it will need to predict unknown data.

4) QUERY SET

The theory of a query set is another meta learning phenomenon. Similar to how the support set was chosen, we will randomly pick data points from dataset D using different data samples. While the support set is used to support the network, data from the query set is used to query the network. While training, the network uses the support information, tries to predict the data from the query set, and then finds the loss to update the network. While testing, data from the testing

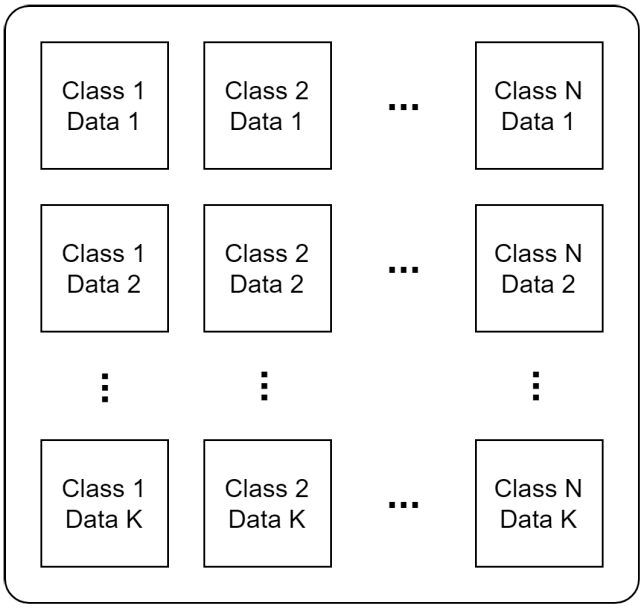


FIGURE 3. Illustrative representation of support and query set.

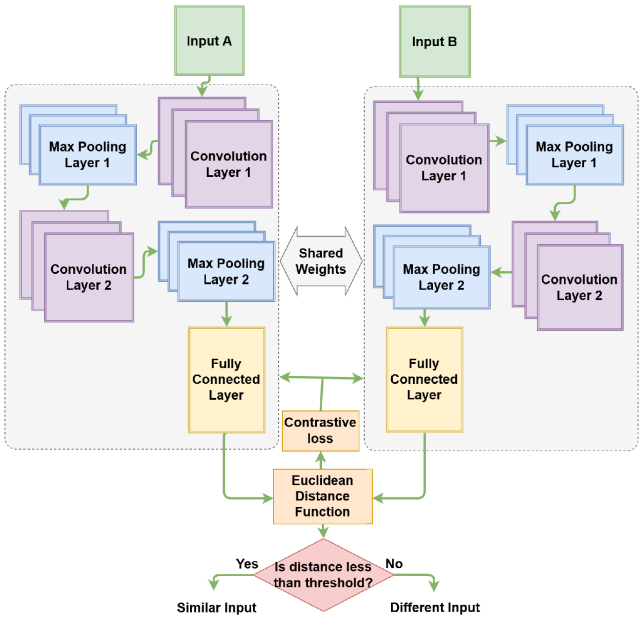


FIGURE 4. Block diagram of the siamese network architecture of our proposed approach.

dataset go into the query set, whereas training data is used for support. The support and query set is visually represented in Fig. 3.

5) SIAMESE NETWORK

Siamese Network is a successor of the meta learning approach. It employs fewer data samples to address problems with insufficient information to train a model. It is one of the most highly sophisticated few-shot learning techniques and consists of two symmetrical neural networks with identical architectures and weights in each network. The prime objective of the Siamese network is to determine whether the two inputs that pass through the two networks are similar or not.

Example: Suppose X_1 and X_2 are two symmetric networks and A_1 and A_2 are two image inputs. We passed them through X_1 and X_2 respectively. The networks will use CNN to extract the features from the images and give embeddings for the inputs. After collecting the embeddings, they will be sent to an Energy Function that uses any distance function to measure the similarity between inputs. If the distance is lower than the threshold value, the inputs belong to the same class and not otherwise.

6) PROTOTYPICAL NETWORK

Another successor of the meta learning technique is the prototypical network. To conduct classification, a normal network tries to learn the metric space. Prototypical networks work on the principle that each class should have a prototypical representation. Each query point should be classified according to its proximity to the class prototype.

Example: Suppose we have three classes $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. Every class has n samples which is represented as $\mathbf{X} = \{X_1, X_2, X_3, \ldots, X_n\}$, $\mathbf{Y} = \{Y_1, Y_2, Y_3, \ldots, Y_n\}$, $\mathbf{Z} = \{Z_1, Z_2, Z_3, \ldots, Z_n\}$. When all the data samples are sent into the network, the network will use CNN to extract the features and get the mean of embedding for each class.

$$X_{prototype} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1}$$

In this way, we will get the average embeddings for every class that is representative of prototype classes like $X_{prototype}$, $Y_{prototype}$ and $Z_{prototype}$. In the testing phase, we will calculate the embedding $P_{embedding}$ for every point P and compute the distance between the class prototype and $P_{embedding}$. Then, we apply softmax to this distance and get the probabilities. We then get the class of query point P from the highest probability.

7) ENERGY FUNCTION

In networks such as the Siamese and Prototypical network, energy functions are used to find the similarity between 2 or more inputs. This allows us to determine the class of the data we are trying to classify and find the loss incurred by the network. While implementing our approach, we used the Euclidean distance function as our energy function. It takes the embeddings of two inputs and finds the distance between them. The general formula of the Euclidean distance function is as follows

$$E(\mathbf{X_1}, \mathbf{X_2}) = ||f(\mathbf{X_1}) - f(\mathbf{X_2})|| \tag{2}$$

where $\mathbf{X_1}$ and $\mathbf{X_2}$ are two feature vectors.

8) PROPOSED APPROACH

Our approach is a hybrid combination of two few-shot learning techniques, the Siamese Network, and the Prototypical

Algorithm 1 Malicious Data Identification Using Siamese Network

Input: Network Flow Images
Output: Malicious Data Identification, Malicious Data
$D_{Train} \leftarrow \{\text{Im}_{Ai}, \text{Im}_{Bj}\}$ where i and j are random indexes for the training set
$D_{Test} \leftarrow \{\text{Im}_{Ai}, \text{Im}_{Bj}\}$ where Im_{Ai} is a random benign sample from training set and j is a random index for testing set
epoch $\leftarrow 0$
threshold $\leftarrow 0.5$

```
while epoch ≤ max_epoch do
    for data in D_Train do
        Embeddings ← f1(data[0], data[1])
        Distance ← D(Embeddings)
        Loss ← Contrastive Loss(Distance, Y_True)
        Update Network parameters using Loss
    end for
end while
for data in D_Test do
    Distance ← Predict(data)
    if Distance ≤ threshold then
        Classify data as Benign
    else
        Classify data as Malicious
    end if
    if Y_true is Malicious then
        D_Malicious ← D_Malicious.append(data[1])
    end if
end for
```

Network. The Siamese network excels at learning the similarity function for two inputs. This helps it to generalize more easily and quickly, allowing it to apply its learning to similar domain data. This makes the network extremely good at binary classification. Additionally, since it trains on the similarity function, it doesn't require a lot of data to provide decent results. However, because the networks only accept two inputs, using them for multi-class classification is quite challenging. This is where the Prototypical network shines. It is also learning the similarity function for the inputs but for multi-class classification settings. As a result, using a combination of Siamese and Prototypical networks allows us to classify a larger number of classes with a small amount of data without sacrificing performance. The following sections describe the architecture of each network.

The Siamese network is used for binary classification to classify the data into benign and malicious. The feature extractor of the Siamese network contains two convolutional and max pool layers, each with 64 filters. The kernel size of the convolutional layers that give us the best outcome is 4×4 with a max pooling size of 2×2. The architecture of our Siamese network is displayed in Fig. 4 for better understanding. We are using euclidean distance as the energy function, which is defined as:

$$D(\mathbf{X_1}, \mathbf{X_2}) = ||f(\mathbf{X_1}) - f(\mathbf{X_2})|| \tag{3}$$

where $\mathbf{X_1}$ and $\mathbf{X_2}$ are data points and $f(\mathbf{X_1})$ and $f(\mathbf{X_2})$ are the embeddings of the data points.

For the loss function, we are using contrastive loss, which is defined as:

$$\text{Contrastive Loss} = Y_{\text{true}} \times D^2 + (1 - Y_{\text{true}}) \times \max(1 - D, 0)^2 \tag{4}$$

The algorithm for the training and testing of the Siamese network is shown in algorithm 1.

The Prototypical network is used for multi-class classification to determine the attack type of the data sample. Its feature extractor also contains two convolutional and max pool layers with 64 filters in each layer. The kernel size for the convolutional layers that give us the best outcome is 4×4 with a max pooling size of 2×2. Fig. 5 illustrates the prototype network architecture for a visual representation. Euclidean distance is again used as the energy function, and negative log probability from the softmax layer is used for the loss calculation. The softmax probability is calculated using the following:

$$P(Y_{\text{true}} = Y_{\text{pred}} \,|\, \mathbf{X}) = \frac{e^{-D(f(\mathbf{X}), \mathbf{X}_{\text{prototype}})}}{\sum_1^n e^{-D(f(\mathbf{X}), \mathbf{X}_{\text{prototype}})}} \tag{5}$$

where $\mathbf{X}$ is the query, $f(\mathbf{X})$ is the embedding of the query and D is the euclidean distance function. Then using P, the loss is

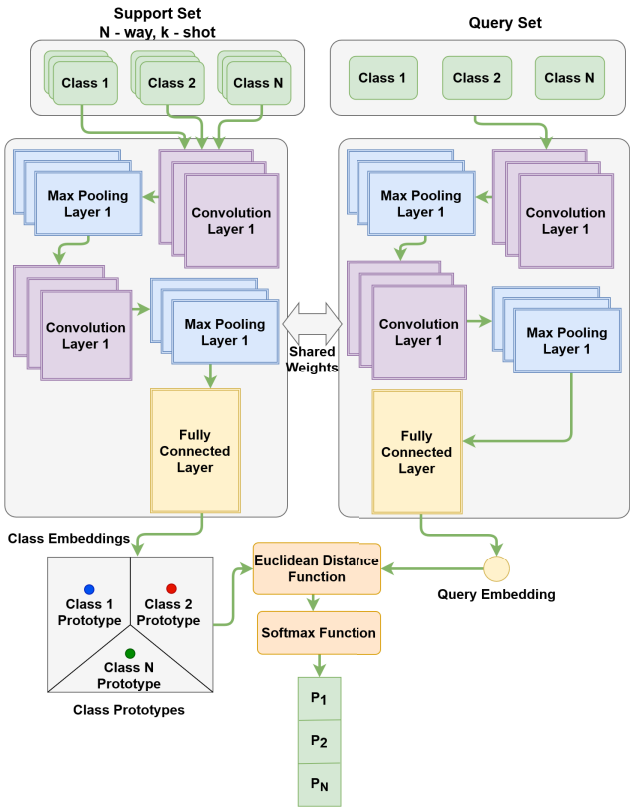


FIGURE 5. Block diagram of the prototypical network architecture of our proposed approach.

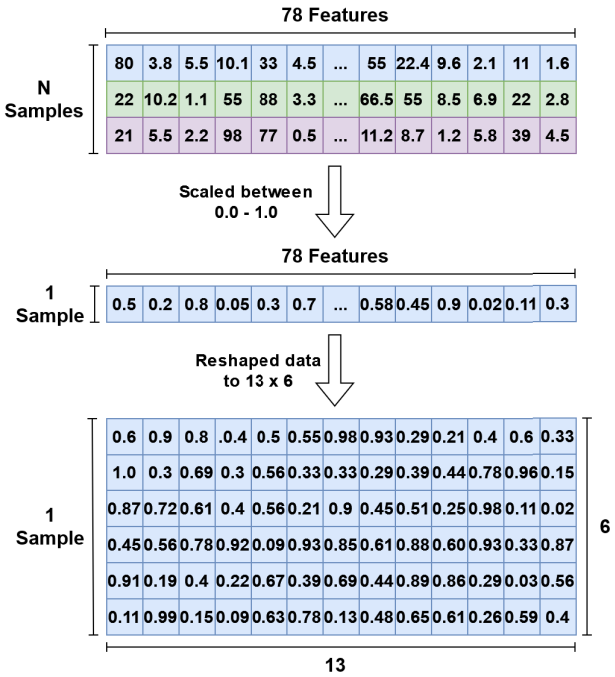


FIGURE 6. Conversion of 1D data from CSE-CIC-IDS2018 to 2D image data.

calculated as follows:

$$\text{Loss} = -\log[P(Y_{\text{true}} = Y_{\text{pred}} \mid \mathbf{X})] \tag{6}$$

The algorithm for the training and testing the Prototypical network is shown in algorithm 2.

C. DATASET

One of the core components of research-based study is the dataset. As we are working on the cyber security domain, a very limited amount of publicly available resources are present. Among them, we have used CSE-CIC-IDS2017 and CSE-CIC-IDS2018. These are the latest publicly available datasets at the time of our work. The main aim of our model is to predict whether network data is malicious or benign correctly. The different labels in the datasets are shown in Table 1.

1) CSE-CIC-IDS2018

This dataset contains 78 features and 15 different labels. The dataset is divided into ten files, each containing benign and a type of attack class, e.g., DDoS, DoS, Bruteforce, and more. The 78 features of the dataset were extracted using CICFlowMeter. The dataset is heavily unbalanced, as evident from the disproportionate distribution between malicious and benign data points. There are 13390249 benign network flow data points, whereas all malicious network flows combined have 2746934 data points.

2) CSE-CIC-IDS2017

This dataset contains nine different files that record data from nine different days. It contains 78 features, just like the CSE-CIC-IDS2018 dataset. However, some of the data labels are different in both datasets. In the CSE-CIC-IDS2018 dataset, they categorize the DDoS attacks into multiple classes, where we found that DDoS attacks were generalized into one class in the CSE-CIC-IDS2017 dataset. Here, we also get portscan and heartbleed, which are absent in the CSE-CIC-IDS2018. The unequal distribution between malicious and benign data points demonstrates the datasets' severe unbalance. There are 1741839 benign network flow data points, whereas all malicious network flow combined have 556556 data points.

3) DATA PREPROCESSING

Data preprocessing is a very crucial step. The model might not provide the desired outcomes because of not having clean and appropriate data. Usually, a dataset contains information that might be irrelevant to the model. In that case, those irrelevant data might lessen the efficiency of that model and increase the time complexity. Different methods use different data preprocessing steps. Our data processing steps are outlined below.

CNNs are renowned for their effectiveness in feature extraction from image-type data. We prioritize CNN for our feature extraction task because it is excellent at extracting features, and we consider the output as our features for further analysis. As a result, we do not manually remove any features. First, we remove all the null and infinity values, then scale the

Algorithm 2 Malicious Data Classification Using Prototypical Network

Input: Network Flow Images, $D_{Malicious}$
Output: Malicious Data Classification
Training Data ← Network Flow images without Benign
epoch ← 0
epoch_size ← *max_size*
  while epoch ≤ max_epoch **do**
    for iter = 0; iter < epoch_size; iter++ **do**
      $Support_{Train}$, $Query_{Train}$ ← Get_Support_Query(training_data, n_way, k_shot)
      Prototype, Embeddings ← $f_2(Support_{Train}, Query_{Train})$
      Distances ← D(Prototype, Embeddings)
      $P(Y = Y_{pred}|X)$ = Softmax(Distances)
      Loss = $-\log(P)$
      Update Network parameters using Loss
    end for
  end while
  max_iter = $D_{Malicious}$.length
  for iter = 0; iter < max_iter; iter++ **do**
    $Support_{Test}$, $Query_{Test}$ ← Get_Test_Support_Query($D_{Malicious}$, training_data, n_way, k_shot)
    Prototype, Embeddings ← $f_2(Support_{Test}, Query_{Test})$
    Distances ← D(Prototype, Embeddings)
    $P(Y = Y_{pred}|X)$ = Softmax(Distances)
    Prediction = max(P)
  end for

TABLE 1. Labels and per label samples in IDS2017 and IDS2018.

Label	CSE-CIC-IDS17	CSE-CIC-IDS18
Benign	1741839	13390249
DDoS	128025	-
DDoS attack-HOIC	-	686012
DDoS attacks-LOIC-HTTP	-	576191
DoS attacks-Hulk	230124	461912
Bot	1956	286191
FTP-BruteForce	7935	193354
SSH-Bruteforce	5897	187589
Infilteration	36	160639
DoS attacks-SlowHTTPTest	5499	139890
DoS attacks-GoldenEye	10293	41508
DoS attacks-Slowloris	5796	10990
DDoS attack-LOIC-UDP	-	1730
Brute Force -Web	1507	611
Brute Force -XSS	652	230
SQL Injection	21	87
PortScan	158804	-
Heartbleed	11	-

data between 0 and 1, and lastly, we convert each data point into image data. The 78 features of the dataset are converted to a shape of 13 × 6.The other shapes that we could have chosen are 39 × 2 and 26 × 3. However, when we tested our models using the different shapes, we found that the other shapes did not provide any performance benefits. As we will be using grayscale images, the final shape of each data point becomes 13 × 6 x 1. Fig. 6 shows the creation of image data.

IV. RESULTS AND DISCUSSIONS

The proposed meta learning technique has been tested on CSE-CIC-IDS2017 and CSE-CIC-IDS2018 datasets. CNN is the feature extraction used to extract the features from the intrusion detection image data and passed to the fully connected layer for classification into the various attack classes.

A. EVALUATION METRICS

The performance of our proposed framework is being evaluated by using some of the well-known indicators such as accuracy, precision, recall, f1 score, and ROC curve. The following indicators are calculated using the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. A true positive is an outcome where the model correctly predicts the positive class. True negative is a result for which the model accurately considers the negative class. A false positive is when the model forecasts the positive class inaccurately. A false negative is where the model forecasts the negative class inaccurately.

Accuracy depicts the measurement of a classification system's overall efficiency. The calculation is as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{7}$$

TABLE 2. Binary classification results.

Dataset	Precision	Recall	F1 Score	Accuracy
IDS2017	94.19	94.13	96.11	94.13
IDS2018	94.67	94.36	93.93	94.36

Among all positives, the portion of data that is correctly classified as positive is called precision. The calculation is as follows:

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

The ratio of accurately classified positives to actual positives is known as recall or sensitivity. The calculation is as follows:

$$Recall = \frac{TP}{(FN + TP)} \tag{9}$$

The f1 score, a benchmark of how well the model performs in classification abilities, is calculated using the harmonic mean of precision and recall. Compared to the standard accuracy metric, the F1 score better represents the classifier's performance. Its value goes from 0 to 1, with 0 representing the lowest possible score and 1 representing the highest possible score. The calculation is as follows:

$$F1\ score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \tag{10}$$

True Positive Rate or TPR provides the proportion of accurate forecasts in predictions of the positive class. Recall is another name for it. The calculation is as follows:

$$TPR = \frac{TP}{(TP + FN)} \tag{11}$$

The false positive rate (FPR) indicates a test's accuracy. It provides the percentage of wrong predictions in the positive class. It is the probability that a false alarm would be triggered. The calculation is as follows:

$$FPR = \frac{FP}{(FP + TN)} \tag{12}$$

ROC curves are a crucial parameter for evaluating the effectiveness of classifiers [1]. They are constructed by plotting the independent true positive rate (TPR) and false positive rate (FPR). The whole test sample's TPR and FPR values are obtained at various threshold settings H in the [0-1] interval to plot the ROC curve. Better performance is shown by classifiers that provide curves that are closer to the top-left corner of the graph.

The area under the ROC curve, also known as the AUC score, is a scalar value that can assess how well the decision model performs overall in classification. A classifier's performance is better as the value gets closer to 1.0, whereas a value closer to 0.5 is on par with guessing labels at random.

B. PERFORMANCE EVALUATION

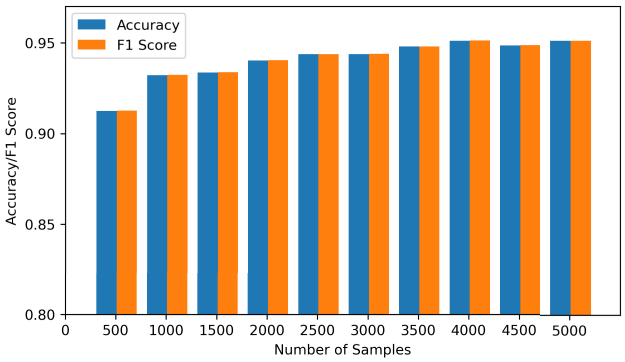
In this subsection, we are mainly focusing on the outcomes of the proposed meta learning model. Here, our model's primary purpose is first to detect if the network packets are malicious and classify them if deemed malicious. To achieve this, we used the CSE-CIC-IDS 2018 dataset, and we also used CSE-CIC-IDS 2017 to illustrate the fact that the outcomes of our model are not dataset-specific. Moreover, we also compare our methods to some of the existing works in this domain.

In our proposed architecture, we randomly selected 3000 samples from our training dataset while balancing the number of samples per class. Then we train our models for multiple epochs in this small sampled dataset and test it on the entire test dataset.

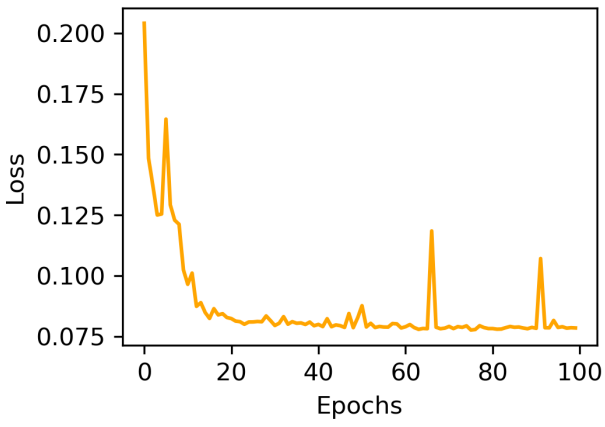
For binary classification using the Siamese network, we found that the overall accuracy of our model is 94.36%. Our model's f1 score, precision, and recall are 93.93%, 94.67%, and 94.36%, respectively. Given that CSE-CIC-IDS 2018 is a very unbalanced dataset, the high f1 score depicts that the unbalanced nature of the dataset is not affecting our model's performance. While using CSE-CIC-IDS 2017, we got an overall accuracy of around 94.13%. Our model's f1 score, precision, and recall with the 2017 dataset are 96.11%, 94.19%, and 94.13%, respectively. This shows that the performance of the proposed approach is consistent irrespective of the dataset. The detailed view of these outcomes is displayed in Table 2.

Moreover, Fig. 7a illustrates the accuracy and f1 score variation of the outcomes for the different number of samples used to train the Siamese network. It always follows an upwards trend, and the f1 scores are close to the accuracy values. It reaches its peak point when the model is trained at around 4000 data samples. However, to be consistent with the hybrid model, we used 3000 data samples to train our Siamese network just like the prototypical network. Using a Siamese network trained over 3000 data samples, we got approximately 94.36% accuracy and 93.93% f1 score. Additionally, a graph is added in Fig. 7b to demonstrate the validation loss of the Siamese network over 100 epochs. The graph shows that the loss seems to stabilize at around 80 to 90 epochs with minor changes, even though there is one anomaly.

Following the Siamese network's results, we incorporated the malicious data into our prototypical network for multiclass categorization. In this experiment, we observed that the accuracy of our model is approximately 90.64% for 14 different labeled malicious network data using CSE-CIC-IDS2018. Moreover, for the IDS2018 dataset, our model's f1 score, precision, and recall score are 91%, 91.66%, and 90.64%, respectively. Among the 14 labels, DDoS LOIC HTTP has the highest f1 score, precision, and recall. We got around 99.97% f1 score, 100% recall, and 99.90% precision for DDoS LOIC HTTP. On the other hand, for SQL Injection, we got a 2.49% f1 score, 53.85% recall, and 1.27% precision. The vast gap in evaluation metrics between DDoS LOIC

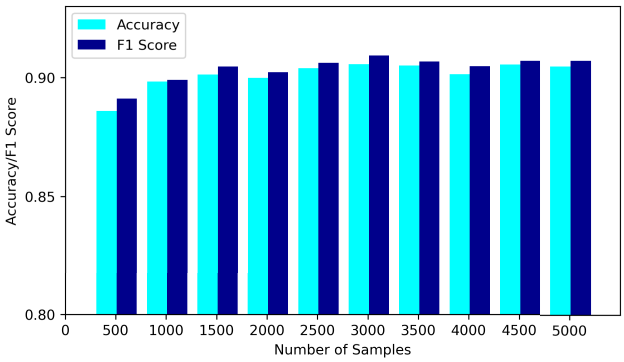


(a) Accuracy and F1 vs No. of Samples of Siamese Network

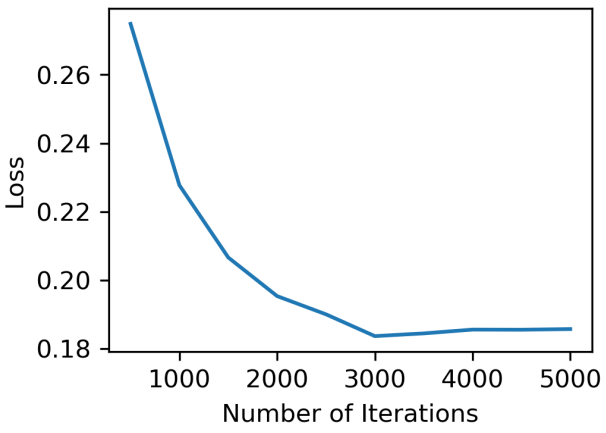


(b) Loss of Siamese network

FIGURE 7. Siamese network evaluation.



(a) Accuracy and F1 vs No. of Samples of Prototypical Network



(b) Loss of Prototypical network

FIGURE 8. Prototypical network evaluation.

HTTP and SQL Injection exists because the number of SQL Injection samples in the IDS2018 dataset is meager compared to others. Therefore, the network had fewer samples to create the prototype for SQL Injection attacks. However, our model works quite well for the labeled data where the number of samples is around 200. For example, using 214 DDoS LOIC UDP data samples, we got around 94.49% precision, 98.85% recall, and 96.62% f1 score.

For a clear visual representation, we added a bar chart shown in Fig. 8a that shows the accuracy and f1 score variation of the prototypical model using the CSE-CIC-IDS2018 dataset. It demonstrates the variation of the results for using different numbers of samples during training. It is seen that both the accuracy and f1 score are following an upwards trend. However, the model has the highest performance when trained with 3000 data samples. Using 3000 data samples, the model accuracy and f1 score are approximately 90% and 91%, respectively. Another graph is shown in Fig. 8b to illustrate the validation loss of the prototypical network over 5000 iterations. The graph shows that the loss seems to stabilize at around 3000 iterations. It is a downward trend graph; the loss is minimal when the model is trained over 3000 iterations.

Furthermore, for the robustness of our proposed architecture, we train our model using CSE-CIC-IDS2017. Using the IDS2017 dataset, we receive approximately 95.68% accuracy for 14 labels. Our model's average f1 score, precision, and recall are 96.1%, 96.5%, and 95.68%, respectively. Among all the labels, we got the highest f1 score, precision, and recall from Heartbleed. However, there were only 11 samples in the total dataset and only four in the testing dataset. So, we do not believe it correctly portrays the model's performance and therefore consider PortScan the best performing label. For PortScan, we got around 99.90% precision, 99.08% recall, and 99.49% f1 score. On the contrary, because of the very small amount of data, we got approximately 1.27% precision, 100% recall, and 2.51% f1 score for SQL Injection data. Table 3 shows the metrics for the individual labels of both CSE-CIC-IDS2018 and CSE-CIC-IDS2017.

As seen from Table 3, our model seems to be struggling with a few attack types, such as Infiltration in CSE-CIC-IDS2017 and SQL Injection in both datasets. We believe the performance is poor in these scenarios because of the meager amount of data available for these attack types. In CSE-CIC-IDS2017, the number of attack samples for Infiltration is 36, which means only 25 (36 * 0.7) data samples are

TABLE 3. Multi-class classification results.

Dataset	Label	Precision	Recall	F1 Score
IDS2017	Bot	0.9191	0.9660	0.9420
	DDoS	0.9121	0.9409	0.9263
	DoS GoldenEye	0.9326	0.9404	0.9365
	DoS Hulk	0.9675	0.9399	0.9535
	DoS Slowhttptest	0.9679	0.9515	0.9597
	DoS Slowloris	0.8404	0.8239	0.8321
	FTP Bruteforce	0.8975	0.9933	0.9430
	Heartbleed	1.0000	1.0000	1.0000
	Infiltration	0.0216	0.8333	0.0422
	PortScan	0.9990	0.9908	0.9949
	SSH Bruteforce	0.9090	0.8360	0.8709
	Bruteforce Web	0.4805	0.3274	0.3895
	SQL Injection	0.0127	1.0000	0.0251
	Bruteforce XSS	0.2338	0.5510	0.3283
IDS2018	Bot	0.9913	0.9896	0.9905
	Brute Web	0.1190	0.6593	0.2017
	Brute XSS	0.0414	0.9706	0.0794
	DDoS HOIC	0.9990	1.0000	0.9995
	DDoS LOIC UDP	0.9449	0.9885	0.9662
	DDoS LOIC HTTP	0.9993	1.0000	0.9997
	DoS GoldenEye	0.9810	0.9640	0.9725
	DoS Hulk	0.9889	0.9971	0.9930
	DoS SlowHTTPTest	0.6024	0.8271	0.6971
	DoS Slowloris	0.8415	0.9982	0.9132
	Brute FTP	0.7523	0.4914	0.5945
	Infiltration	0.9882	0.8892	0.9361
	SQL Injection	0.0127	0.5385	0.0249
	Brute SSH	0.9946	1.0000	0.9973

used to train the model. In CSE-CIC-IDS2017, there are only 21 data samples for SQL Injection and 87 for CSE-CIC-IDS2018. Therefore, only 14 (21 * 0.7) and 60 (87 * 0.7) SQL Injection data samples are used to train the model using CSE-CIC-IDS2017 and CSE-CIC-IDS2018, respectively. Thus, whereas most other classes are being trained on with around 200 samples, Infiltration and SQL injection was trained on much fewer samples, which can explain the performance gap.

Another observation from the results is that our model also seems to struggle with Bruteforce Web and Bruteforce XSS even though they have been trained with 200 samples. Analysis of the confusion matrix in Fig. 9 showed us that the prototypical network confuses the two classes. The confusion matrix shows that most of the misclassified Bruteforce Web are Bruteforce XSS, and most Bruteforce XSS are Bruteforce Web. From this, we can infer that the attack patterns of these

TABLE 4. Evaluation of siamese network with different kernel and filter sizes.

Dataset	Kernel	Filter Size	Accuracy	F1 Score
IDS2017	2 x 2	32	90.588	93.986
		64	91.488	94.331
		128	90.425	93.415
	3 x 3	32	91.719	94.473
		64	93.222	95.506
		128	93.952	95.925
	4 x 4	32	90.374	93.401
		64	94.131	96.113
		128	93.665	95.724
IDS2018	2 x 2	32	94.241	93.027
		64	94.306	93.912
		128	94.336	93.899
	3 x 3	32	94.103	93.209
		64	93.431	92.995
		128	93.560	93.149
	4 x 4	32	93.774	93.248
		64	94.357	93.929
		128	93.651	93.104

classes are similar, and thus their prototypes will also have a smaller distance between them. This can explain why the prototypical network is struggling with these classes.

Hyperparameter tuning is essential to choose appropriate parameters to help the model better fit the data. Table 4 represents the accuracy and F-1 score for the different combinations of the Siamese network's kernel size and filter size. From Table 4, it is visible that our Siamese network achieved the best outcomes when the kernel size is 4×4 and the filter size is 64 for both datasets.

The accuracy and F-1 score for the various kernel size and filter size combinations of the prototypical network are shown in the Table 5. Table 5 makes it quite evident that our prototypical network performed best for both datasets when the kernel size is 4×4 and the filter size is 64.

By analyzing the above mentioned information, we choose the combination of 4×4 kernel size and 64 filter size for our architecture as we get the best outcomes with this combination.

As mentioned above, the ROC curve is an important metric to determine the performance of classifiers. Thus we are using the ROC curve to see the performance of our proposed architecture. Fig. 10 shows how our Siamese network performs during malicious data identification. The figure shows that the performance of the Siamese network does not vary significantly between the datasets. Additionally, the AUC score referenced in Table 6 is close to 1.0, signifying that the model performs well. However, it works better for the CSE-CIC-IDS2017 dataset as the curve is more to the top left corner of the graph. This is also shown in the AUC score.

ROC curves are primarily used to evaluate the effectiveness of binary classifiers. However, we can consider two distinct

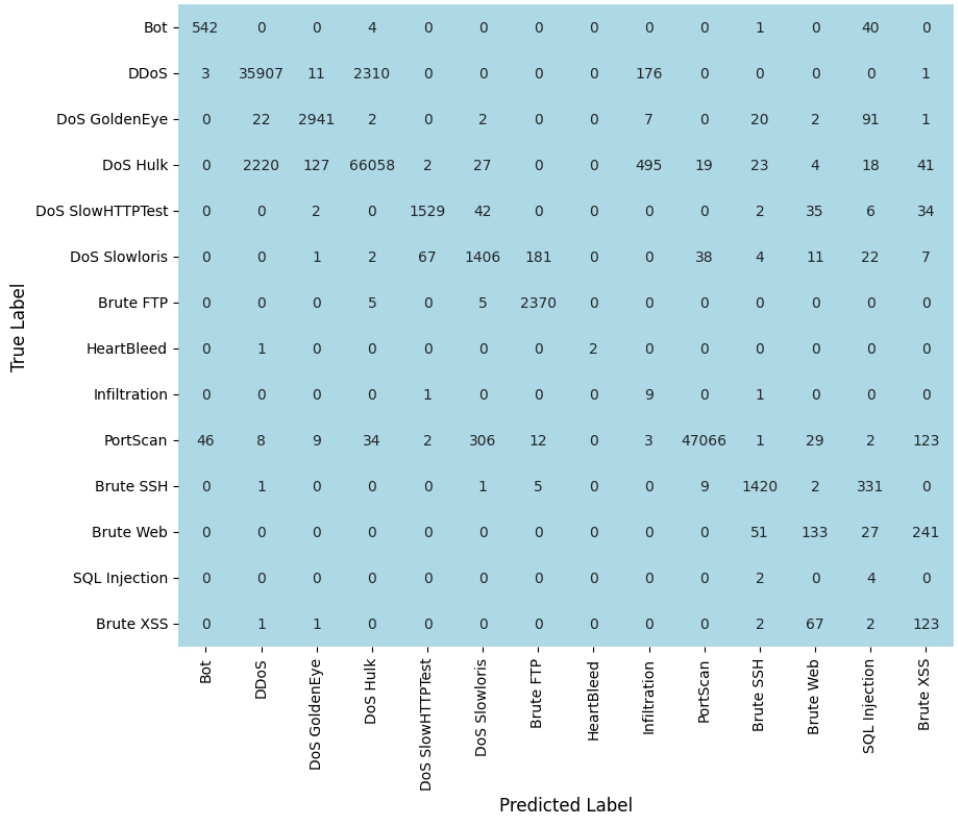


FIGURE 9. Confusion matrix for CSE-CIC-IDS17.

approaches to multi-class classification: One vs. Rest and One vs. One. We are considering One vs. Rest to measure the performance of our Prototypical network. At a time, one label is considered positive, and the rest are considered negative. Using this, TPR and FPR are calculated, and the ROC curve is plotted. To measure the overall performance of the Prototypical network, we also provide the micro and macro averages of all the ROC curves.

ROC curves for CSE-CIC-IDS17 and CSE-CIC-IDS18 are displayed in Fig. 11a and Fig. 11b, respectively. In Fig. 11a, it is observed that the performance of the label Infiltration is worse compared to the rest of them, as evident from the figure. It is also visible in the AUC score in Table 7.

In Fig. 11b, it is seen that the performance of SlowHTTPTest and Bruteforce FTP are worse compared to others as both the curves of these two labels get closer to the top of the graph. Moreover, it is also observable in the AUC score in Table 7, which is calculated from the ROC curve.

C. COMPARATIVE STUDY

The following details will demonstrate an overview between our work and some of the current work in this domain. Table 8 also shows this comparison.

Meta learning techniques may be utilized to train models efficiently with limited data; we, therefore, incorporated a hybrid meta deep learning strategy. We trained our models using 3000 data samples, applying the settings mentioned in tables 4 and 5. Here, we could classify 15 different labels in the CSE-CIC-IDS2018 with about 94% and

TABLE 5. Evaluation of prototypical network with different kernel and filter sizes.

Dataset	Kernel	Filter Size	Accuracy	F1 Score
IDS2017	2 x 2	32	86.795	88.561
		64	91.605	92.337
		128	92.664	93.383
	3 x 3	32	95.019	95.415
		64	95.486	95.996
		128	95.169	95.801
	4 x 4	32	94.974	95.521
		64	95.683	96.105
		128	95.509	95.833
IDS2018	2 x 2	32	87.631	88.116
		64	88.405	88.855
		128	88.455	88.975
	3 x 3	32	89.640	89.975
		64	90.257	90.464
		128	90.162	90.453
	4 x 4	32	90.033	90.303
		64	90.640	91.001
		128	90.495	90.751

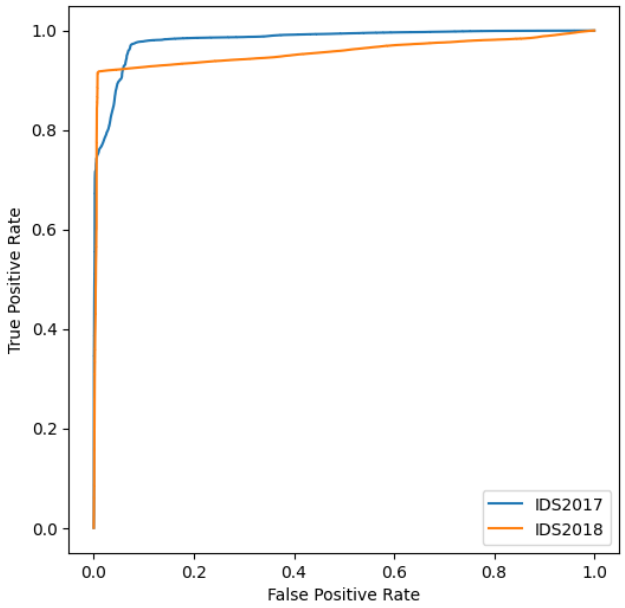


FIGURE 10. ROC curves of siamese network for CSE-CIC-IDS2017 and CSE-CIC-IDS2018.

TABLE 6. AUC scores of siamese network of different datasets.

Dataset	AUC
CSE-CIC-IDS2017	98.119
CSE-CIC-IDS2018	95.575

90% accuracy using Siamese and prototypical networks, respectively, despite using 3000 data samples.

In [8], the authors also conducted an experiment on binary and multiclass classification for DoS attacks. They employed a CNN for this task. They trained and tested their model using about 10407862 samples of data from the CSE-CIC-IDS2018, and their binary classification accuracy was 91.5%. For six distinct labeled DoS attacks, they also achieved 91.5% for multiclass classification. For comparison purposes, We used their method for 14 malicious attacks from the CSE-CIC-IDS 2018 dataset. For multiclass identification, their approach achieved an accuracy of about 91.69% and 90.70% f1 score, 92.34% precision, and 91.69% recall.

On the other hand, with only 3000 data, we reached an accuracy of 90.29%. The f1 score, precision, and also recall closely match their results. They used 10407862 samples to arrive at a conclusion, which our model achieved with 3000 samples. Our approach even outperformed them in most cases. For example, classifying infiltration using our hybrid meta learning approach, we get 98.82% precision, 88.92% recall, and 93.61% f1 score, whereas using the approach taken by [8], we get 97.72% precision, 40.30% recall, and 57.07% f1 score.

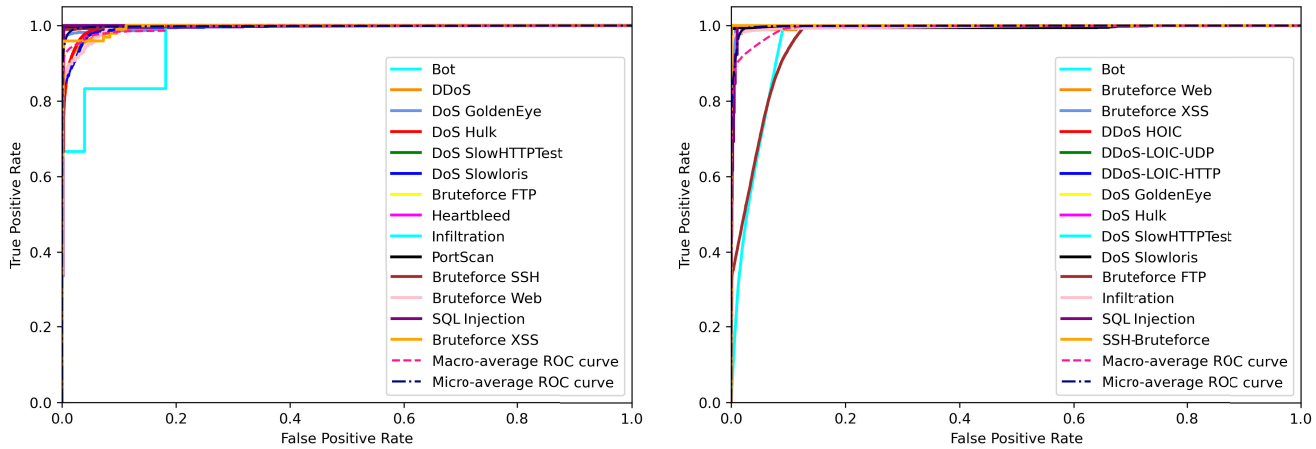
We also used the ROC curve and AUC score to compare the outcomes of this study [8] with our work. The ROC curve provides the best evaluation picture for classification

TABLE 7. OvR AUC scores of prototypical network of different datasets.

Dataset	Label	OvR AUC
CSE-CIC-IDS2017	Bot	0.999
	DDoS	0.993
	DoS GoldenEye	0.997
	DoS Hulk	0.994
	DoS SlowHTTPTest	0.999
	DoS Slowloris	0.993
	Bruteforce FTP	1.000
	Heartbleed	1.000
	Infiltration	0.963
	PortScan	0.999
	Bruteforce SSH	0.998
	Bruteforce Web	0.994
	SQL Injection	0.998
	Bruteforce XSS	0.995
	Macro Average	0.995
	Micro Average	0.998
CSE-CIC-IDS2018	Bot	1.000
	Bruteforce Web	0.997
	Bruteforce XSS	0.998
	DDoS HOIC	1.000
	DDoS-LOIC-UDP	1.000
	DDoS-LOIC-HTTP	1.000
	DoS GoldenEye	1.000
	DoS Hulk	0.999
	DoS SlowHTTPTest	0.965
	DoS Slowloris	0.996
	Bruteforce FTP	0.967
	Infiltration	0.997
	SQL Injection	0.998
	SSH-Bruteforce	1.000
	Macro Average	0.994
	Micro Average	0.998

problems. Fig. 12 shows OvR ROC curves using the approach taken by [8] for 14 malicious classes. It is clear from the figure that their Infiltration performance is much worse compared to other attack types. From Table 9, it is visible that the AUC score of infiltration using [8]'s approach is around 85% using CSE-CIC-IDS2018. However, from Table 9, it is evident that our proposed framework works better as, in our model, the AUC score for infiltration is 99.7%. Moreover, the other labels that perform worse are still above 95% score. The AUC score for a few attack samples is worse because those specific attack patterns vary within themselves, making them not easily distinguishable. Our model might require more samples of those attack types to understand their similarities better. From this analysis, we can say that our work performs better regarding the AUC score, which is also evident from the micro and macro average scores.

In [9], authors recommended a Siamese Network using a one-shot learning mechanism to classify cyber attacks. In total, they used 274729 samples for their experiment. Among them, 248607 samples are normal data, and the rest are divided into DoS(Hulk), DoS(Slowloris), FTP Brute Force, and SSH Brute Force. For five different cyber attacks,



(a) One vs. Rest ROC Curves of Prototypical Network using CSE-CIC-IDS2017

(b) One vs. Rest ROC Curves of Prototypical Network using CSE-CIC-IDS2018

FIGURE 11. ROC curves of prototypical network for CSE-CIC-IDS2017 and CSE-CIC-IDS2018.

TABLE 8. Multi-class classification comparison with related work.

Related Work	Dataset	Data Used	No. of Classes	Accuracy	Precision	Recall	F1 Score
Our Hybrid Meta Approach	IDS2017	3000	14	95.68	96.50	95.68	96.10
	IDS2018	3000	14	90.64	91.66	90.64	91.00
CNN Based [8]	IDS2018	10407862	6	91.50	60.00	84.75	84.75
Siamese One Shot [9]	IDS2017	274729	5	80.81 - 82.50	83.06	82.50	82.50
Siamese Capsule [13]	IDS2017	3025	8	95.25	98.37	96.29	97.32
SPN [25]	IDS2017	369878	8	92.00	93.78	92.44	92.48
FC-Net [18]	IDS2017	10000	5	95.39	-	-	-

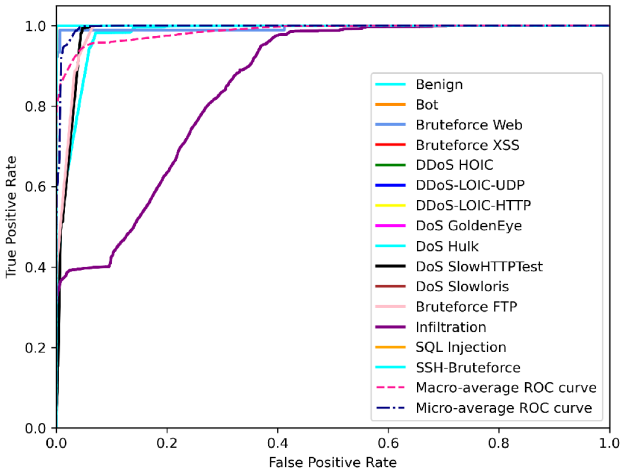


FIGURE 12. ROC curves of CNN based approach [8] with 15 Labels.

they achieved 80.81% to 82.5% accuracy using the CSE-CIC-IDS 2017 dataset. On the other hand, we considered all the attacks and utilized the same dataset and achieved an accuracy of about 94.22% for 14 distinct attack labels with 3000 data samples.

Authors proposed a Siamese Network based model in [13] to enhance the network intrusion detection using unbalanced training data. They presented their experiment with CSE-CIC-IDS 2017. They achieved an accuracy of about 95.25% for eight different classes. However, our suggested system achieved accuracy that is very close to their results while considering all forms of malicious data.

The authors in [25] proposed a few shot traffic classification method where they used Siamese Prototypical Network to classify network data. In their approach, they used the architecture of Siamese Network to build a Prototypical Network to classify 8 distinct malicious classes. From the comparison that [25] showed between their method and baseline methods, it can be seen that their method achieved 92.48% F1 score, 92.44% Recall, 93.78% Precision and approximately 92% accuracy using dataset CSC-CIC-IDS2017. Our proposed method achieved 95.68% accuracy using only 3000 data samples from the dataset CSC-CIC-IDS2017 for 14 distinct classes which is demonstrated in Table 8.

A few shot network based intrusion detection model named FC-Net was proposed by [18] where they used data of only 5 different attack classes in CSE-CIC-IDS2017 dataset. They performed binary classification between each attack type and normal type to prove the effectiveness of their approach. On average, their approach was able to achieve 95.39% accuracy using 10000 data samples. Proposed approach used

TABLE 9. Comparison with OvR AUC scores of CNN based approach [8] with 14 labels.

Label	OvR AUC of [8]	Proposed Approach
Bot	0.999	1.000
Bruteforce Web	0.995	0.997
Bruteforce XSS	0.999	0.998
DDoS HOIC	1.000	1.000
DDoS-LOIC-UDP	1.000	1.000
DDoS-LOIC-HTTP	1.000	1.000
DoS GoldenEye	0.999	1.000
DoS Hulk	0.999	0.999
DoS SlowHTTPTest	0.984	0.965
DoS Slowloris	0.999	0.996
Bruteforce FTP	0.986	0.967
Infiltration	0.855	0.997
SQL Injection	0.999	0.998
SSH-Bruteforce	0.999	1.000
Macro Average	0.987	0.994
Micro Average	0.996	0.998

far fewer samples from the entire dataset, and was able to perform multi-class classification on 14 classes, achieving an overall accuracy of 95.68% on the same dataset.

V. CONCLUSION

In this paper, we introduced a hybrid meta learning approach to detect and identify malicious packet data using multi-class classification. This study aimed to achieve a highly effective model that can classify malicious network data with very few samples. The advancement of meta learning technologies can be an excellent solution in providing a trustworthy mechanism. The ability of Siamese and Prototypical networks to rapidly fit the data allowed us to address the issue of malicious data insufficiency and perform well even with highly unbalanced datasets. This approach ensures a secure transmission by detecting malicious packet data and classifying them into multiple classes. This information can be used for more fine-grained control over which preventative measure to take for the specific attack type. Our proposed architecture can be trained without many data samples. Hybrid meta learning techniques can train the model efficiently with minimal data for most of the attack types compared to the other existing approaches. Thus, it minimizes the problem of data insufficiency. Furthermore, our model can classify 15 distinct classes with greater than 90% accuracy using only 3000 data samples. Therefore, we can say that the outcomes of our research has a strong substantial impact on the research field. As it fills the gap in the existing research field where a model can be trained with a very minimal amount of data and yet detect and classify the malicious classes efficiently to ensure a secure interaction between users.

However, even though the model can be trained with small amounts of data, we still need data for the new attack types and to improve the performance of the proposed model. To further improve the accuracy of our model, we can increase the number of samples used to train the model by removing the restriction of using a smaller subset of the whole dataset. Oversampling techniques such as SMOTE can also be incorporated in this regard. If necessary, we can also generalize the attack types to improve the accuracy of the model. Additionally, one of the limitations of the proposed model is that it depends on a centralized data repository. To address this, we want to incorporate federated learning as our future task, which will provide real time data and the proposed model will be trained with diverse decentralized data samples. We believe it will improve the performance of our model to detect more malicious attacks efficiently. Furthermore, incorporating attention mechanisms with the proposed framework might give direction to future researchers in this domain. Most importantly, we anticipated that the findings of our strategy might be an essential component of a platform for end-to-end trust networking. In conclusion, we sincerely believe that our efforts will be beneficial in solving the security crisis in the upcoming days.

REFERENCES

[1] C. E. Metz, "Basic principles of ROC analysis," *Seminars Nucl. Med.*, vol. 8, no. 4, pp. 283–298, Oct. 1978.

[2] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020.

[3] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, "Low-rate DoS attacks, detection, defense, and challenges: A survey," *IEEE Access*, vol. 8, pp. 43920–43943, 2020.

[4] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, and T.-H. Kim, "Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey," *IEEE Access*, vol. 10, pp. 121173–121192, 2022.

[5] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102266–102291, 2022.

[6] M. Wazid, A. K. Das, S. Shetty, P. Gope, and J. J. P. C. Rodrigues, "Security in 5G-enabled Internet of Things communication: Issues, challenges, and future research roadmap," *IEEE Access*, vol. 9, pp. 4466–4489, 2021.

[7] R. Bie, X. Jin, C. Chen, C. Xu, and R. Huang, "Meta learning intrusion detection in real time network," in *Proc. Int. Conf. Artif. Neural Netw.*, 2007, pp. 809–816.

[8] 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.

[9] H. Hindy, C. Tachtatzis, R. Atkinson, D. Brosset, M. Bures, I. Andonovic, C. Michie, and X. Bellekens, "Leveraging Siamese networks for one-shot intrusion detection model," *J. Intell. Inf. Syst.*, vol. 60, no. 2, pp. 407–436, Apr. 2023.

[10] J. Wang, Z. Zhu, J. Li, and J. Li, "Attention based Siamese networks for few-shot learning," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Nov. 2018, pp. 551–554, doi: 10.1109/ICSESS.2018.8663732.

[11] J. Kim, Y. Shin, and E. Choi, "An intrusion detection model based on a convolutional neural network," *J. Multimedia Inf. Syst.*, vol. 6, no. 4, pp. 165–172, Dec. 2019.

[12] D. Park, S. Kim, H. Kwon, D. Shin, and D. Shin, "Host-based intrusion detection model using Siamese network," *IEEE Access*, vol. 9, pp. 76614–76623, 2021.

[13] Z.-M. Wang, J.-Y. Tian, J. Qin, H. Fang, and L.-M. Chen, "A few-shot learning-based Siamese capsule network for intrusion detection with imbalanced training data," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–17, Sep. 2021.

[14] J. Wang and Y. Zhai, "Prototypical Siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 178–181.
[15] G. Cheng, L. Cai, C. Lang, X. Yao, J. Chen, L. Guo, and J. Han, "SPNet: Siamese-prototype network for few-shot remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 3099033.
[16] J. F. Cañola Garcia and G. E. T. Blandon, "A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks," *IEEE Access*, vol. 10, pp. 83043–83060, 2022, doi: 10.1109/ACCESS.2022.3196642.
[17] J. Verma, A. Bhandari, and G. Singh, "A meta-analysis of role of network intrusion detection systems in confronting network attacks," in *Proc. 8th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, New Delhi, India, Mar. 2021, pp. 506–511.
[18] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3540–3552, 2020, doi: 10.1109/TIFS.2020.2991876.
[19] S. Mo, Z. Sun, and C. Li, "Siamese prototypical contrastive learning," 2022, *arXiv:2208.08819*.
[20] T. Wang, Q. Lv, B. Hu, and D. Sun, "A few-shot class-incremental learning approach for intrusion detection," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Athens, Greece, Jul. 2021, pp. 1–8, doi: 10.1109/ICCCN52240.2021.9522260.
[21] K. Wu, P. Wang, and Z. Wang, "Few-shot malicious traffic classification based on Siamese neural network," in *Proc. IEEE 23rd Int. Conf High Perform. Comput. Commun.; 7th Int. Conf Data Sci. Syst.; 19th Int. Conf Smart City; 7th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Hainan, China, Dec. 2021, pp. 1670–1675, doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00246.
[22] W. Lu and Y. Ding, "A network malicious traffic detection method based on semi-supervised deep learning," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2021, pp. 1–6, doi: 10.1109/ICSPCC52875.2021.9564717.
[23] M. Ishaque, M. G. M. Johar, A. Khatibi, and M. Yamin, "Intrusion detection system using binary and multiclass deep neural network classification," in *Proc. 9th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, New Delhi, India, Mar. 2022, pp. 749–753, doi: 10.23919/INDIACom54597.2022.9763122.
[24] H. Xu and Y. Wang, "A continual few-shot learning method via meta-learning for intrusion detection," in *Proc. IEEE 4th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Dali, China, Oct. 2022, pp. 1188–1194, doi: 10.1109/ICCASIT55263.2022.9986665.
[25] G. Miao, G. Wu, Z. Zhang, Y. Tong, and B. Lu, "SPN: A method of few-shot traffic classification with Out-of-Distribution detection based on Siamese prototypical network," *IEEE Access*, vol. 11, pp. 114403–114414, 2023, doi: 10.1109/ACCESS.2023.3325065.
[26] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, Jan. 2022, doi: 10.1109/JIOT.2021.3084796.
[27] C. A. Fadhilla, M. D. Alfikri, and R. Kaliski, "Lightweight meta-learning BotNet attack detection," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8455–8466, May 2023, doi: 10.1109/JIOT.2022.3229463.
[28] C. Rong, G. Gou, C. Hou, Z. Li, G. Xiong, and L. Guo, "UMVD-FSL: Unseen malware variants detection using few-shot learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Shenzhen, China, Jul. 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533759.

SAKIB UDDIN TAPU received the B.Sc. degree in computer science and engineering from BRAC University. He is currently a Cloud Engineer in Bangladesh. His research interests include cloud computing, cloud security, information security, intrusion detection systems, and meta-deep learning.

SAMIRA AFRIN ALAM SHOPNIL received the B.Sc. degree in computer science and engineering from BRAC University. She is currently a Software Engineer in Bangladesh. Her research interests include information security, artificial intelligence, and meta-deep learning.

RABEYA BOSRI TAMANNA received the B.Sc. degree in computer science and engineering from BRAC University. Her research interests include data-driven prediction systems, artificial intelligence, machine learning, deep learning, software engineering, large-scale data mining, cleaning, and inference.

M. ALI AKBER DEWAN (Member, IEEE) received the B.Sc. degree in computer science and engineering from Khulna University, Bangladesh, in 2003, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2009. From 2003 to 2008, he was a Lecturer with the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh, where he was an Assistant Professor, in 2009. From 2009 to 2012, he was a Postdoctoral Researcher with Concordia University, Montreal, QC, Canada. From 2012 to 2014, he was a Research Associate with École de Technologie Supérieure, Montreal. He is currently an Associate Professor with the School of Computing and Information Systems (AU), Athabasca University (AU), Athabasca, AB, Canada, where served as the Chair of the department from 2019 to 2022. His research interests include artificial intelligence in education, affective computing, computer vision, data mining, information visualization, machine learning, and medical image analysis. His research works are supported by Alberta Innovates, Mitacs, and Natural Science and Engineering Research Council of Canada. He has served as an editorial board member, the chair/the co-chair, and a TPC member for several prestigious journals and conferences. He received the Dean's Award and the Excellent Research Achievement Award for his excellent academic performance and research achievements during his Ph.D. studies in South Korea.

MD. GOLAM RABIUL ALAM (Member, IEEE) received the B.S. degree in computer science and engineering and the M.S. degree in information technology, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2017. He was a Postdoctoral Researcher with the Department of Computer Science and Engineering, Kyung Hee University, from March 2017 to February 2018. He is currently a Full Professor of computer science and engineering with the Department of Computer Science and Engineering, BRAC University, Bangladesh. His research interests include healthcare informatics, mobile cloud and edge computing, ambient intelligence, and persuasive technology. He is a member of IEEE IES, CES, CS, SPS, CIS, KIISE, and IEEE ComSoc. He received several best paper awards at prestigious conferences.