Contrastive Learning for Fraud Detection from Noisy Labels

Vinay M.S.
University of Arkansas
Fayetteville, AR 72701, USA
vmadanbh@uark.edu

Shuhan Yuan
Utah State University
Logan, UT 84322, USA
Shuhan.Yuan@usu.edu

Xintao Wu
University of Arkansas
Fayetteville, AR 72701, USA
xintaowu@uark.edu

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00117

***Abstract*—Detecting frauds in computing platforms involves identifying malicious user activity sessions. Recently, deep learning models have been employed to design fraud detection approaches. Effective training of these deep learning models requires a large amount of well-annotated sessions. However, due to the cost of expert annotation, many organizations rely on heuristics to perform automated annotation, which leads to the noisy label learning problem. It is well known that the performance of deep learning models can easily degrade because of noisy or inaccurate labels. To tackle this challenge, we propose a supervised Contrastive Learning based Fraud Detection (CLFD) framework, which is designed to operate in the noisy label setting. CLFD employs an effective label corrector for correcting noisy labels and which is specifically designed for the fraud detection task. Then, by employing the corrected labels, it trains a fraud detector through supervised contrastive learning, and derives separable representations. We empirically evaluate our CLFD framework and other state-of-the-art baselines on benchmark datasets. Our CLFD framework demonstrates superior performance over state-of-the-art baselines.**

***Index Terms*—fraud detection; contrastive learning; noisy label; label correction.**

I. Introduction

Computing platforms such as cloud computing systems, usually experience a large volume of malicious or fraudulent activities due to the anonymity and openness character of the Internet. In order to protect the legitimate users, it is extremely important to identify such malicious activities. In practice, the user activities are usually modeled as an activity session. For example, in a computer system, an activity session is a sequence of activities starting with system log-in and ending with system log-out. Recently, many deep learning models have been proposed in the literature [1] for detecting malicious sessions. These models generate session representations by making normal sessions deviate from the malicious ones in the representation space for deriving anomaly scores.

The two main challenges in the fraud detection task are dataset imbalance and session diversity [1]. *Dataset imbalance:* In the ground-truth, only a few malicious sessions are recorded, which leads to extreme dataset imbalance. *Session diversity:* It is well known that user activity sessions and especially malicious sessions, usually exhibit high diversity [1]. The malicious users could design various attacks, which leads to high session diversity. Recently, a few deep learning based fraud detection approaches have been proposed in the literature [2]–[4] which specifically address the dataset imbalance and session diversity challenges. Specifically, Vinay et al. [4] proposed a supervised contrastive learning based fraud detection framework, where the supervised contrastive learning approach [5] extends the vanilla self-supervised contrastive learning approach to the supervised setting. The main goal here is to push samples from the same class closer and contrast with other class samples in the representation space. Due to this class-specific clustering effect in the representation space, we can effectively address both session diversity and dataset imbalance challenges in the fraud detection task [4].

One limitation of the supervised contrastive learning approach is that it relies on well-labeled samples for training. However, in many real-world fraud detection scenarios, due to the high costs of expert annotations, many financially constrained organizations find it difficult to hire such experts for manually annotating the recorded sessions [6]. In such scenarios, organizations rely on historic security rules or heuristics to perform automated annotations, leading to the noisy label data [6]. However, the contemporary fraud detection approaches [2]–[4] have not been designed to operate in the noisy label setting. Specifically, due to the effect of noisy supervision, the performance of a supervised contrastive learning based model can easily degrade [7].

To address the limitations of supervised contrastive learning for the noisy label learning task, Li et al. [8] and Yi et al. [9] proposed label correction approaches that correct the noisy labels by employing sample similarity analysis, and employ these corrected labels to train the model through supervised contrastive learning. However, these approaches are specifically designed for the image data, and assume that the samples belonging to the same class have considerable shared features. Hence, these approaches are not suitable for the fraud detection task due to the presence of the session diversity challenge. Note that even corrected labels are not accurate and have uncertainties [10]. As a consequence, even with label correction, the performance of the supervised contrastive learning model can still get degraded. The existing supervised contrastive learning based noisy label learning approaches [8], [9] do not address this label correction uncertainty challenge.

To address these challenges, we propose a supervised Contrastive Learning based Fraud Detection (CLFD) framework which operates in the noisy label setting. To correct the noisy

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00117

labels, CLFD employs a label corrector which is designed by suitably adapting the fraud detection framework called *CLDet*, which was proposed by Vinay et al. [3]. Specifically, CLDet employs the self-supervised contrastive learning model to learn session representations. Then, it trains a classifier over the learned session representations by employing the noise sensitive cross entropy loss. CLDet is designed to specifically address the dataset imbalance challenge. Unlike the supervised contrastive learning model, the session representations learned from the self-supervised contrastive learning model are not influenced by the presence of noisy labels, and can also aid the noisy label learning task [7], [11], [12]. Hence, we leverage this CLDet framework to design our label corrector. Specifically, we train the classifier in CLDet by our proposed mixup version of the noise robust Generalized Cross Entropy (GCE) [13] loss, instead of the original noise sensitive cross entropy loss. Note that the self-supervised contrastive learning model pushes a session and its corresponding augmented versions closer and contrasts with other sessions in the representation space. It does not specifically induce the class-specific clustering effect which is achieved by the supervised contrastive learning model. Therefore, the supervised contrastive learning model provides a better opportunity to address the session diversity challenge. However, the supervised contrastive learning model can under-perform in the noisy label setting and requires an effective label corrector to guide the model supervision [7]. Hence, CLFD first employs the label corrector to correct the noisy labels, and further adopts the corrected labels to train a fraud detector for detecting malicious sessions. Moreover, we propose a weighted supervised contrastive loss to effectively address the label correction uncertainty challenge, where the uncertainties associated with the corrected labels are used to weigh the corresponding sessions inside the supervised contrastive loss. We further enhance the noise robustness of our fraud detector by training a separate classifier over the learned session representations with our proposed mixup GCE loss. We summarize our main contributions below:

- We propose a supervised contrastive learning based fraud detection framework called CLFD, which is specifically designed to operate in the noisy label setting.
- We propose a weighted supervised contrastive loss which is designed to address the challenge of uncertainty in the label correction process. We theoretically show that this weighted supervised contrastive loss is upper bounded by the ideal loss. Additionally, we propose the mixup version of the GCE loss for training classifiers under the noisy label setting, and theoretically show its efficacy.
- We present an empirical study on three benchmark fraud detection datasets: CERT [14], UMD-Wikipedia [15], and Open-stack [16], in which we show the superior performance of our CLFD framework over state-of-the-art baselines.

II. Related Work

Learning Under Noisy Labels. There is a large body of work presented in the literature for the noisy label learning task. We direct the interested readers to [17] for a comprehensive survey on the different proposed approaches. Some of the recent and popular noisy label learning approaches are: robust loss functions [13], [18]–[24] which propose noise robust losses, noise transition matrix [25]–[28] which requires the knowledge of class-specific noise rates, sample selection [24], [29], [30] which selects supposedly clean labeled samples based on the sample loss analysis, and label correction [10], [31]–[36] which corrects the given noisy labels and further trains the employed model by using these corrected labels. Specifically, the label correction approaches have outperformed the remaining approaches [7]. Hence, we have designed our framework by employing an effective label corrector. However, the existing label correction approaches have been designed for image datasets. In our empirical study, we select some of recently-proposed label correction approaches [10], [31] as baselines, and show that they under-perform on the fraud detection task. Zhang et al. [37] proposed the mixup data augmentation strategy which has been used in many recent noisy label learning approaches [17]. There is no work in the literature which has theoretically studied benefits of the mixup GCE loss. Zhao et al. [6] developed an anomaly detection framework under the noisy label setting for image datasets. Specifically, they assume that each sample has multiple noisy labels, and train a mixture-of-experts model to learn from multiple labels. However, in our work, we do not employ multi-label setting.

Recently, both Li et al. [8] and Yi et al. [9] have employed supervised contrastive learning for the noisy label learning task on image datasets. Specifically, Li et al. [8] perform label correction through the nearest neighbor method. Then, confident samples are selected based on the agreement between corrected and given labels. By using these confident samples, confident pairs are selected based on sample similarity analysis. Finally, through these confident pairs, a supervised contrastive learning model is trained. Yi et al. [9] also select confident pairs based on sample similarity analysis. However, they propose a novel contrastive regularization function to learn sample representations over noisy labels where the label noise does not dominate. In our empirical study, we select both these approaches [8], [9] as baselines and show that they fail to provide noticeable results on the fraud detection task due to the session diversity challenge. Jaiswal et al. [38] presented a comprehensive survey on the applications of self-supervised contrastive learning on computer vision and NLP domains. In the literature, there is no work studying the benefits of supervised contrastive learning for the fraud detection task under the noisy label setting.

Insider Threat Detection. It refers to detecting frauds committed by organizational insiders. Deep learning based approaches [39]–[45] are popularly employed in detecting insider threats. Recently, many deep learning based approaches [2]–

[4], [46] have been specifically designed to operate on imbalanced datasets. However, all these existing deep learning based insider threat detection approaches do not operate under the noisy label setting. In our empirical study, we select some of the recent deep learning based insider threat detection approaches [2], [3] as baselines, and show that they underperform under the noisy label setting.

Log Anomaly Detection. The goal here is to detect anomalies occurring in the computing system log data [16]. Interested readers can refer to [47] for a comprehensive survey on log anomaly detection approaches. Recent approaches employ deep learning models to detect log anomalies [48], [49]. However, these approaches are not specifically designed for the noisy label learning task.

III. Proposed CLFD Framework

The activities performed by a user are modeled through activity sessions. Specifically, each session can consist of T user activities. Each activity in the session is represented as an embedding vector that is trained via the word-to-vector model. Let $\mathbf{x}_{it}$ denote the word-to-vector representation of the t^{th} activity of the i^{th} session. Here, $\mathbf{x}_i = \{\mathbf{x}_{it}\}_{t=1}^{T}$ represents the *raw representation* of the i^{th} session. Let $\mathcal{Y} = \{0, 1\}$ denote the label space where 0 and 1 denote normal and malicious sessions, respectively. Let $\widetilde{y}_i$ and y_i denote noisy and ground truth label of $\mathbf{x}_i$, respectively. We do not assume the availability of any clean labeled sessions. The available noisy training set is denoted as $\widetilde{\mathcal{T}} = \widetilde{\mathcal{T}}^0 \cup \widetilde{\mathcal{T}}^1$, where $\widetilde{\mathcal{T}}^0$ and $\widetilde{\mathcal{T}}^1$ denote the set of noisy normal and malicious sessions in $\widetilde{\mathcal{T}}$, respectively. Let $(\mathbf{x}_i, \widetilde{y}_i) \in \widetilde{\mathcal{T}}$ denote the i^{th} training sample. In our problem setting, we deal with the commonly used noises in the literature [13] which are the uniform and class-dependent label noises. For the uniform noise, the noise rate is denoted as $\eta = P(\widetilde{y}_i \neq y_i)$. Similarly, for the class dependent noise, we denote $\eta_{10} = P(\widetilde{y}_i = 0 | y_i = 1)$ and $\eta_{01} = P(\widetilde{y}_i = 1 | y_i = 0)$. Our framework architecture is shown in Figure 1. There are two main components in our framework: *label corrector* and *fraud detector*. We describe both these components below.

A. Label Corrector

We employ our trained label corrector to predict the class for each session in $\widetilde{\mathcal{T}}$ and use these predicted classes as corrected labels to train our fraud detector. Our proposed label corrector architecture is shown in Figure 1b. We design our label corrector by suitably adapting the CLDet fraud detection framework proposed by Vinay et al. [3]. Our label corrector has two main components: self-supervised pre-training component and a classifier. The self-supervised pre-training component generates session representations, which are trained by the vanilla self-supervised SIMCLR contrastive loss [50]. After this training, the session representations are used as inputs to the classifier, which is trained by our proposed noise robust mixup GCE loss. The major modification that we make in the CLDet framework in order to design our label corrector is that, we train the classifier by our proposed mixup GCE loss instead of the original noise sensitive cross entropy loss.

1) Mixup GCE Loss: The popular cross entropy loss suffers from model over-fitting issue when applied on the noisy label learning task [13]. Additionally, the cross entropy loss is an unbounded loss which amplifies the model over-fitting issue. To address these issues, Zhang et al [13] proposed the GCE loss. However, the vanilla GCE loss still faces the issue of label memorization effect [51]. The label memorization effect means the model is overconfident about the relationship between the input features and their corresponding labels, which is problematic under the noisy label setting as these labels could be incorrect. Recently, mixup based data augmentation strategy proposed by Zhang et al. [37] effectively addresses this label memorization issue. Specifically, augmented samples are generated through the randomized interpolation of the sample features and their corresponding labels. Hence, to increase the noise robustness of vanilla GCE loss, we propose the mixup version of GCE loss for the fraud detection task[1].

Let $\mathbf{v}_i$ denote the session representation generated by the self-supervised pre-training component[2] for $\mathbf{x}_i$. Let f denote the classifier function. The classifier softmax output vector for $\mathbf{v}_i$ is denoted as $f(\mathbf{v}_i) = [f_0(\mathbf{v}_i), f_1(\mathbf{v}_i)]^\top$, where $f_0(\mathbf{v}_i)$ and $f_1(\mathbf{v}_i)$ denote softmax probabilities for normal and malicious session classes, respectively. Let $q \in (0, 1]$ and $\widetilde{\mathbf{e}}_i = [\widetilde{e}_{i0}, \widetilde{e}_{i1}]^\top$ denotes the noisy one-hot encoded label of $\mathbf{x}_i$. The vanilla GCE loss [13] is given by:

$$l_{GCE}(f(\mathbf{v}_i), \widetilde{\mathbf{e}}_i) = \sum_{k=0}^{1} \frac{\widetilde{e}_{ik}}{q} \left(1 - f_k(\mathbf{v}_i)^q\right) \tag{1}$$

We propose our mixup strategy below which is designed by leveraging the mixup strategy presented by Zhang et al. [37]. Let $\left(\mathbf{v}_i^\lambda, \widetilde{\mathbf{m}}_i\right)$ denote the *mixup sample* corresponding to $(\mathbf{x}_i, \widetilde{\mathbf{e}}_i)$. Here, $\mathbf{v}_i^\lambda = \lambda \mathbf{v}_i + (1-\lambda)\mathbf{v}_j$, $\widetilde{\mathbf{m}}_i = \lambda \widetilde{\mathbf{e}}_i + (1-\lambda)\widetilde{\mathbf{e}}_j$, $\lambda \sim Beta(\beta, \beta)$, $\lambda \in [0, 1]$, and the session $\mathbf{x}_j \in \widetilde{\mathcal{T}}$ is sampled from the opposite noisy class to which $\mathbf{x}_i$ belongs $(\widetilde{y}_j \neq \widetilde{y}_i)$. For our proposed mixup strategy, the mixup version of GCE loss is given by:

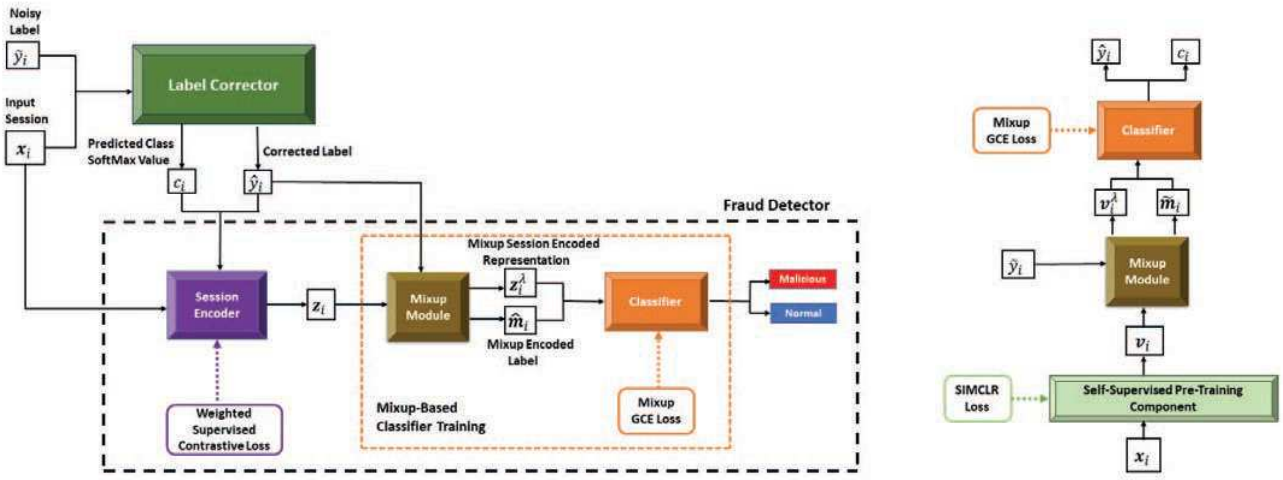
$$l_{GCE}^{\lambda}(f(\mathbf{v}_i^\lambda), \widetilde{\mathbf{m}}_i) = \sum_{k=0}^{1} \frac{\widetilde{m}_{ik}}{q} \left(1 - f_k(\mathbf{v}_i^\lambda)^q\right) \tag{2}$$

We construct a training batch $S = \{\mathbf{x}_i\}_{i=1}^{R}$ by randomly sampling R sessions from $\widetilde{\mathcal{T}}$. We train the classifier by calculating the batch loss for each batch S, which is given by:

$$\mathcal{L}_{GCE}^{\lambda} = \frac{1}{R} \sum_{\mathbf{x}_i \in S} l_{GCE}^{\lambda}(f(\mathbf{v}_i^\lambda), \widetilde{\mathbf{m}}_i) \tag{3}$$

[1]As a starting point, we have proposed the mixup version of GCE loss and in our future work, we will analyze other available noise robust loss functions.

[2]A detailed description of the CLDet framework architecture including the procedure to derive session representations, employed SIMCLR contrastive loss, and training procedure is available in [3].



(a) Main framework

(b) Label corrector expanded

Fig. 1: Illustration of our CLFD framework architecture. Label corrector is employed to correct the noisy labels. By using these corrected labels, the fraud detector is trained and deployed for inference.

Theoretical Analysis of $\mathcal{L}^{\lambda}_{GCE}$. Zhang et al. [13] presented a comprehensive theoretical analysis study on vanilla GCE loss shown in Equation 1. We extend some of their theoretical results to our proposed mixup GCE loss shown in Equation 2. We show the noise robustness property of $\mathcal{L}^{\lambda}_{GCE}$ by analyzing its gradient. Let $\boldsymbol{\phi}$ denote the set of parameters for the classifier. The gradient of $\mathcal{L}^{\lambda}_{GCE}$ w.r.t $\phi_j \in \boldsymbol{\phi}$ is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}^{\lambda}_{GCE}}{\partial \phi_j} &= \frac{\partial}{\partial \phi_j}\left[\frac{1}{R}\sum_{\mathbf{x}_i \in S}\sum_{k=0}^{1}\frac{\widetilde{m}_{ik}}{q}\left(1-f_k(\mathbf{v}_i^{\lambda})^q\right)\right] \\ &= -\frac{1}{R}\sum_{\mathbf{x}_i \in S}\sum_{k=0}^{1}\widetilde{m}_{ik} f_k(\mathbf{v}_i^{\lambda})^{q-1}\frac{\partial f_k(\mathbf{v}_i^{\lambda})}{\partial \phi_j} \\ &= -\frac{1}{R}\sum_{\mathbf{x}_i \in S}\sum_{k=0}^{1} w_{ik}\frac{\partial f_k(\mathbf{v}_i^{\lambda})}{\partial \phi_j} \end{aligned} \tag{4}$$

Here, the loss gradient weight $w_{ik} = \widetilde{m}_{ik} f_k(\mathbf{v}_i^{\lambda})^{q-1}$. The two main challenges in the noisy label learning task are model over-fitting and label memorization.

Model Over-Fitting. During the training stage of a classifier, if the label of a training sample is inaccurate then usually, the classifier softmax outputs for the training sample have weak agreements with the given inaccurate one-hot encoded label [13]. In such scenarios, noise sensitive losses such as cross entropy give greater emphasis to such weak agreement samples. As a consequence, the model learns by over-fitting to such weak agreement samples. Consider the gradient of $\mathcal{L}^{\lambda}_{GCE}$ shown in Equation 4. Suppose, the target $\widetilde{m}_{ik}$ and the classifier prediction $f_k(\mathbf{v}_i^{\lambda})$ have weak agreement between them. Then clearly, w_{ik} will be closer to zero. Hence, less emphasis will be placed on such weak agreement samples during the learning stage, and $\mathcal{L}^{\lambda}_{GCE}$ avoids over-fitting the weak agreement samples.

Label Memorization. Deep learning models are prone to label memorization. At higher noise rates, label memorization effect can lead to poor decision boundaries even with noise robust losses such as the vanilla GCE loss [51]. During each training epoch, for each sample $(\mathbf{x}_i, \widetilde{\mathbf{e}}_i) \in \widetilde{\mathcal{T}}$, we construct its corresponding randomly interpolated sample $\left(\mathbf{v}_i^{\lambda}, \widetilde{\mathbf{m}}_i\right)$ through our proposed mixup strategy. As a consequence, we effectively address the label memorization challenge through $\mathcal{L}^{\lambda}_{GCE}$.

The unhinged/Mean Absolute Error (MAE) loss is a noise robust loss. However, it has a slow rate of optimization convergence [13]. We can define the mixup version of unhinged/MAE loss as: $l^{\lambda}_{MAE}(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i) = \sum_{k=0}^{1}\widetilde{m}_{ik}\left(1-f_k(\mathbf{v}_i^{\lambda})\right)$. It is trivial to see that when $q=1$, our mixup GCE loss becomes the mixup unhinged/MAE loss.

The Categorical Cross Entropy (CCE) loss can achieve fast rate of optimization convergence. However, it is sensitive to the label noise [13]. The mixup version of CCE loss is given by: $l^{\lambda}_{CCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right) = -\sum_{k=0}^{1}\widetilde{m}_{ik}\log\left(f_k(\mathbf{v}_i^{\lambda})\right)$. Theorem 1 states that when $q \to 0$, $l^{\lambda}_{GCE}(\cdot,\cdot)$ converges to $l^{\lambda}_{CCE}(\cdot,\cdot)$ and $l^{\lambda}_{GCE}(\cdot,\cdot)$ can achieve a high rate of optimization convergence by maintaining noise robustness property. We outline all proofs of our theoretical results in Section VI.

Theorem 1. $\lim_{q\to 0} l^{\lambda}_{GCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right) = l^{\lambda}_{CCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right)$

Next, we aim to establish both upper and lower bounds for $l^{\lambda}_{GCE}(\cdot,\cdot)$. Note that unbounded losses such as cross entropy loss are typically extremely sensitive to noise. In some cases, this cross entropy loss could become large when the noisy label mismatches the model prediction. As a consequence, the model would attempt to counteract the large loss by

over-fitting the label noise, leading to poor generalization performance [13]. However, $l^{\lambda}_{GCE}(\cdot,\cdot)$ has clearly defined bounds as shown in Theorem 2.

Theorem 2. $\min(\lambda, 1-\lambda)\frac{2-2^{(1-q)}}{q} \leq l^{\lambda}_{GCE}(f(\mathbf{v}^{\lambda}_i), \widetilde{\mathbf{m}}_i) \leq \frac{1}{q}$

Now we will analyze the classifier risk associated with $l^{\lambda}_{GCE}(\cdot,\cdot)$ for noisy labels by comparing it with ground truth label risk. Specifically, we show that classifier risks associated with $l^{\lambda}_{GCE}(\cdot,\cdot)$ for both uniform and class dependent noise rates are not substantial by upper bounding these risks with the corresponding ground truth label risks. Let $\mathcal{D}$ denote the noisy training set distribution. We denote the risk for f w.r.t $l^{\lambda}_{GCE}(\cdot,\cdot)$ for ground truth labels as $R^{\lambda}_{GCE}(f) = \mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}}\left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \mathbf{m}_i\right)\right]$. Here, $\mathbf{m}_i = \lambda\mathbf{e}_i + (1-\lambda)\mathbf{e}_j$ denotes the ground truth mixup encoded label for $\mathbf{v}^{\lambda}_i$, and $y_j \neq y_i$. Similarly, we denote the risk for f for both uniform and class conditional noise rates as $\widetilde{R}^{\lambda}_{GCE}(f) = \mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}}\left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right)\right]$. Theorem 3 states that under the uniform noise setting, the noisy risk $\widetilde{R}^{\lambda}_{GCE}(f)$ is upper bounded by the ground truth risk $R^{\lambda}_{GCE}(f)$.

Theorem 3. For the uniform label noise rate η, we have that: $\widetilde{R}^{\lambda}_{GCE}(f) \leq R^{\lambda}_{GCE}(f) + \frac{\eta}{q}$

Let $\widetilde{\tau}^0 = P(\widetilde{y}_i = 0)$ and $\widetilde{\tau}^1 = P(\widetilde{y}_i = 1)$. The ground truth class conditional risks for f are denoted as: $R^{\lambda}_{GCE}(f|y_i=1) = \mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}}\left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \mathbf{m}_i\right)|y_i=1\right]$ and $R^{\lambda}_{GCE}(f|y_i=0) = \mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}}\left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \mathbf{m}_i\right)|y_i=0\right]$. Theorem 4 states that under the class conditional noise setting, the noisy risk $\widetilde{R}^{\lambda}_{GCE}(f)$ is upper bounded by the ground truth class conditional risks.

Theorem 4. For the class dependent label noise rates η_{01} and η_{10}, we have that:

$$\widetilde{R}^{\lambda}_{GCE}(f) \leq \widetilde{\tau}^1\left(R^{\lambda}_{GCE}(f|y_i=1) + \frac{\eta_{10}}{q}\right) + \widetilde{\tau}^0\left(R^{\lambda}_{GCE}(f|y_i=0) + \frac{\eta_{01}}{q}\right)$$

B. Fraud Detector

The main goal of the fraud detector is to learn to identify malicious sessions through the supervision from our trained label corrector. We denote the corrected label and the corresponding one-hot encoding of $\mathbf{x}_i$ as $\widehat{y}_i$ and $\widehat{\mathbf{e}}_i$, respectively. We employ two stage training for our fraud detector: supervised pre-training and mixup-based classifier training. The reason is that, for the noisy label learning task, Li et al [8] recommend training a classifier over representations learned by a supervised contrastive learning model by using a noise robust loss.

1) Supervised Pre-Training: Let c_i denote the output softmax value (confidence or posterior probability) for the predicted/corrected class of $\mathbf{x}_i$, which is provided by our trained label corrector. Here, $c_i = \max\left[f_0(\mathbf{v}_i), f_1(\mathbf{v}_i)\right]^{\top}$. We employ c_i as a weighting parameter in the supervised contrastive loss. Initially, by employing our trained label corrector, we generate corrected labels for all sessions $\mathbf{x}_i \in \widetilde{\mathcal{T}}$. Let $\widehat{\mathcal{T}}^1$ denote the set of those sessions in $\widetilde{\mathcal{T}}$ that have been predicted as malicious by the label corrector. We employ a separate encoder network for our fraud detector, which maps a session from its raw representation $\mathbf{x}_i$ to an *encoded representation* vector $\mathbf{z}_i$. We adopt LSTM as the foundation of our encoder to derive the encoded session representations. Our encoder has two hidden layers with the same dimensions. We derive $\mathbf{z}_i$ by averaging the LSTM final hidden layer representations. We construct a training batch $S = \{\mathbf{x}_i\}^{R}_{i=1}$ from $\widetilde{\mathcal{T}}$. Since our framework is specifically designed to operate on imbalanced training data, in-order to effectively contrast corrected malicious and normal sessions, for each training batch S, we create a corresponding auxiliary batch $S^1 = \{\mathbf{x}^1_i\}^{M}_{i=1}$, by randomly sampling M corrected malicious sessions from $\widehat{\mathcal{T}}^1$. We propose a weighted supervised contrastive loss which is designed by leveraging the supervised contrastive loss presented by Khosla et al. [5]. This loss is given by:

$$\mathcal{L}_{Sup} = \frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \quad (5)$$

Here, the set $A(\mathbf{x}_i) = \left(S \cup S^1\right) - \{\mathbf{x}_i\}$, and the set $B(\mathbf{x}_i)$ contains sessions $\mathbf{x}_p \in A(\mathbf{x}_i)$ such that both $\mathbf{x}_i$ and $\mathbf{x}_p$ share the same corrected label. We employ $c_i c_p$ as a weight for the session pair $(\mathbf{x}_i, \mathbf{x}_p)$. Let α denote the temperature parameter. The individual loss for the pair $(\mathbf{x}_i, \mathbf{x}_p)$ is defined as:

$$l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) = -\log\left(\frac{exp(\cos\left(\mathbf{z}_i\cdot\mathbf{z}_p\right)/\alpha)}{\sum_{\mathbf{x}_j\in A(\mathbf{x}_i)} exp(\cos\left(\mathbf{z}_i\cdot\mathbf{z}_j\right)/\alpha)}\right) \quad (6)$$

2) Mixup-Based Classifier Training: In the mixup-based classifier training stage, we employ a Fully Connected Neural Network (FCNN) having two layers as a separate classifier for our fraud detector. Specifically, the first layer is an input layer which receives the encoded session representation $\mathbf{z}_i$ as input. It is equipped with a Leaky ReLU activation function. The second layer is an output/classification layer. It is equipped with a softmax activation function. We employ $l^{\lambda}_{GCE}(\cdot,\cdot)$ to train our FCNN. We employ the corrected labels obtained from our label corrector for the training supervision. We use this trained FCNN for our test case inference. The training procedure for our fraud detector is outlined in Algorithm 1.

Time Complexity Analysis. The CLFD training cost involves the training costs of our label corrector and fraud detector. The training cost of the self-supervised contrastive learning based label corrector is upper bounded by the training cost of the supervised contrastive learning based fraud detector [5]. Hence, we analyze the time complexity of our fraud detector training procedure. Specifically, we analyze the forward pass in training, and the number of times the pair loss $l_{Sup}(\cdot,\cdot)$ is invoked. This time complexity is given by: $O\left(|\widetilde{\mathcal{T}}|\,(R+M)\right)$.

Theoretical Analysis of $\mathcal{L}_{Sup}$. We show the effect of our proposed loss on robust session representation learning by

Algorithm 1 Training procedure for the fraud detector.

Inputs: $\widetilde{\mathcal{T}} = \widetilde{\mathcal{T}}^1 \cup \widetilde{\mathcal{T}}^0$, R, M, β, trained label corrector, and our fraud detector.
Output: well trained fraud detector.
obtain corrected labels for all sessions in $\widetilde{\mathcal{T}}$ from the trained label corrector;
construct $\widehat{\mathcal{T}}^1 = \left\{\mathbf{x}_i \in \widetilde{\mathcal{T}} | \widehat{y}_i = 1\right\}$;
[**Supervised Pre-Training**]
generate training batches from $\widetilde{\mathcal{T}}$;
for each training batch $S = \{\mathbf{x}_i\}_{i=1}^{R}$ **do**
  create the auxiliary batch $S^1 = \{\mathbf{x}_i^1\}_{i=1}^{M}$ from $\widehat{\mathcal{T}}^1$;
  obtain $\widehat{y}_i$ and c_i for each session $\mathbf{x}_i \in S \cup S^1$ from the trained label corrector;
  for each session $\mathbf{x}_i \in S$ **do**
    construct $A(\mathbf{x}_i) = \left(S \cup S^1\right) - \{\mathbf{x}_i\}$;
    construct $B(\mathbf{x}_i) = \{\mathbf{x}_p \in A(\mathbf{x}_i) | \widehat{y}_p = \widehat{y}_i\}$;
    for each session $\mathbf{x}_p \in B(\mathbf{x}_i)$ **do**
      calculate $l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right)$ by using Eq 6
  calculate $\mathcal{L}_{Sup}$ by using Eq 5 and train the session encoder;
[**Mixup-Based Classifier Training**]
for each training batch $S = \{\mathbf{x}_i\}_{i=1}^{R}$ **do**
  for each session $\mathbf{x}_i \in S$ **do**
    sample a session $\mathbf{x}_j$ from $\widetilde{\mathcal{T}}$ such that $\widehat{y}_j \neq \widehat{y}_i$;
    sample $\lambda \sim Beta(\beta, \beta)$;
    construct $\mathbf{z}_i^{\lambda} = \lambda \mathbf{z}_i + (1-\lambda)\mathbf{z}_j$ and $\widehat{\mathbf{m}}_i = \lambda \widehat{\mathbf{e}}_i + (1-\lambda)\widehat{\mathbf{e}}_j$;
    calculate $l_{GCE}^{\lambda}(f(\mathbf{z}_i^{\lambda}), \widehat{\mathbf{m}}_i)$ by using Eq 2;
  calculate $\mathcal{L}_{GCE}^{\lambda}$ by using Eq 3 and train the FCNN;
return the well trained fraud detector;

analyzing its gradient. Let $\boldsymbol{\theta}$ denote the set of parameters for the session encoder. The gradient of $\mathcal{L}_{Sup}$ w.r.t to $\theta_j \in \boldsymbol{\theta}$ is given by:

$$\frac{\partial \mathcal{L}_{Sup}}{\partial \theta_j} = \frac{1}{R}\sum_{i=1}^{R} \frac{1}{|B(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in B(\mathbf{x}_i)} (c_i c_p) \frac{\partial l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right)}{\partial \theta_j} \quad (7)$$

The gradient $\frac{\partial \mathcal{L}_{Sup}}{\partial \theta_j}$ is weighted by the term $c_i c_p$. Through supervised contrastive learning, our session encoder learns to push the encoded session representations of $\mathbf{x}_i$ and $\mathbf{x}_p$ closer in the encoded representation space. Here, $\mathbf{x}_i$ and $\mathbf{x}_p$ have been predicted to belong to the same class by the label corrector. However, these predicted labels have uncertainties. We require a mechanism to reduce the learning effect from those session pairs $(\mathbf{x}_i, \mathbf{x}_p)$ which are predicted with low confidence (output softmax value closer to 0.5). Hence, we employ $c_i c_p$ as a weighting parameter in $\mathcal{L}_{Sup}$ to achieve this goal.

For our theoretical analysis, we employ a hypothetical oracle supervised contrastive loss. We assume that the ground truth label (y) for a session $\mathbf{x} \sim D$, can be obtained by giving $\mathbf{x}$ as input to the oracle. This oracle supervised contrastive loss expressed in terms of expectation is given by:

$$\mathcal{L}_{Orc} = \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[\frac{1}{|\breve{B}(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in \breve{B}(\mathbf{x}_i)} l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \quad (8)$$

Here, $\breve{B}(\mathbf{x}_i)$ contains sessions $\mathbf{x}_p \in A(\mathbf{x}_i)$ such that both $\mathbf{x}_i$ and $\mathbf{x}_p$ share the same ground truth label. Let c denote the corrected class confidence (output softmax value) for a session $\mathbf{x} \sim \mathcal{D}$, which is obtained through our trained label corrector. Here, $P(c \approx 1)$ denotes the probability that the label corrector is highly confident, and $c_p \not\approx 1$ denotes that c_p is not closer to 1. We can express $\mathcal{L}_{Sup}$ in-terms of expectation as:

$$\mathcal{L}_{Sup} = \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[\frac{1}{|B(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in B(\mathbf{x}_i)} (c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \quad (9)$$

Theorem 5.

$$\mathcal{L}_{Sup} \leq P(c \approx 1) \Bigg\{ P(c \approx 1)\mathcal{L}_{Orc} + \mathop{E}_{\substack{\mathbf{x}_i \sim \mathcal{D} | c_i \approx 1 \\ c_p \not\approx 1}} \left[(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \Bigg\} + \mathop{E}_{\mathbf{x}_i \sim \mathcal{D} | c_i \not\approx 1} \left[(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right]$$

Theorem 5 states that $\mathcal{L}_{Orc}$ upper-bounds $\mathcal{L}_{Sup}$. We further analyze the effect of our $\mathcal{L}_{Sup}$ on robust session representation learning by comparing it with other possible variants. Specifically, we consider the unweighted version of $\mathcal{L}_{Sup}$, and session filtering based supervised contrastive loss which discards a session pair having low joint confidence. We theoretically show the merits of our $\mathcal{L}_{Sup}$ against the other loss variants. We provide these additional theoretical underpinnings on $\mathcal{L}_{Sup}$ in Section VII.

IV. Experiments

We describe our experimental setup including datasets and baselines used in this paper and then discuss our empirical analysis results which includes label corrector performance, training latency, and ablation analysis results.

A. Experimental Setup

1) Datasets: We use three benchmark fraud detection datasets for our empirical study: CERT [14], UMD-Wikipedia [15], and OpenStack [16].

CERT [14]. The CERT dataset is a benchmark dataset for insider threat detection. There are 48 malicious and 1,581,358 normal sessions. The insider sessions are recorded chronologically over 516 days. To avoid extreme training latency, we randomly sample 10,000 normal sessions from the first 460 days and include them in our training set. Similarly, we randomly sample 500 normal sessions from 461 to 516 days to construct our test set. For the malicious sessions, we randomly sample 30 malicious sessions, and include them in our training

set. The remaining 18 malicious sessions are included in our test set.

UMD-Wikipedia [15]. This dataset records the activity sessions of users who have edited the Wikipedia website. In this dataset, there are 5486 normal and 4627 malicious sessions. We randomly sample 1000 normal sessions to construct our test set and include all the remaining 4486 normal sessions in our training set. To simulate the training dataset imbalance scenario, we randomly sample 80 malicious sessions and include them in our training set. From the remaining malicious sessions, we randomly sample 500 malicious sessions, and include them in our test set.

OpenStack [16]. This dataset records the activity sessions of users who have used the OpenStack cloud services. In this dataset, there are 244,908 normal and 18,434 malicious sessions. We randomly sample 10,000 and 1000 normal sessions and include them in our training and test sets, respectively. Similarly, we randomly sample 60 and 100 malicious sessions, and include them in our training and test sets, respectively.

2) Training Details: To effectively train our label corrector and fraud detector, we set the number of dimensions of the activity and session representations, and the hidden layer size of our LSTM based session encoder to 50. To avoid extreme memory requirements during encoder training, we opt for medium sized training batches. Specifically, we use 100 sessions (R) in each training batch. We employ the *session reordering* based augmentation strategy proposed by Vinay et al. [3] for the self-supervised pre-training of our label corrector. Specifically, for each session, we randomly select an activity sub-sequence of length 3, and reorder activities in this sub-sequence. The temperature parameter α shown in Equation 6 is set to its default value 1. For simulating the uniform noise rate η, we randomly flip the ground truth label of a session with a probability η [13]. Similarly, for simulating the class conditional noise rates η_{10} and η_{01}, we randomly flip the ground truth label of a malicious and a normal session with probabilities η_{10} and η_{01}, respectively [52]. Since we are operating on an extremely imbalanced training set, we constrain the noise rates to be within 0.5 so that a few accurately labeled malicious sessions are available for model training. In real world scenarios, if the dataset noise rate can be estimated then, for noise rates above 0.5, we can easily invert the noisy labels, and again bring back the new noise rates within 0.5. We empirically analyze the performance of our CLFD framework by using different values for the uniform noise rate η. For the class-dependent noise rates, we set $\eta_{10} = 0.3$, and $\eta_{01} = 0.45$. The GCE loss parameter q is set to 0.7 as recommended by Zhang et al. [13]. For the mixup hyper-parameter β, Zhang et al. [37] recommend that mixup interpolation should have sufficient strength to prevent the label memorization effect. Hence, we set β to 16. We set the size of the malicious session auxiliary batch (M) used in the training of our session encoder to 20. We use the Adam optimizer [53] with a learning rate of 0.005 and we use 10 training epochs for both self-supervised and supervised pre-training of our label corrector and fraud detector, respectively. For the mixup-based classifier training in our label corrector and fraud detector, we employ 500 epochs. We utilize three metrics to measure the fraud detection performance: F_1, False Positive Rate (FPR), and Area Under the Receiver Operating Characteristics Curve (AUC-ROC). We report the mean and standard deviation of performance scores after 5 times of running.

3) Baselines: We compare our framework with eight state-of-the-art baselines: DivMix [31], ULC [10], Sel-CL [8], CTRR [9], Few-Shot [2], CLDet [3], DeepLog [16], and LogBert [48]. Specifically, DivMix, ULC, Sel-CL, and CTRR have been designed for the noisy label learning task. DivMix and ULC employ co-teaching based approach while, Sel-CL and CTRR employ supervised contrastive learning based approach. These noisy label learning approaches have been designed to originally operate on image datasets and employ neural networks for image data such as ResNet-18 [31]. Hence, we cannot directly apply these baselines for our fraud detection task which operates on sequential data. We suitably replace their neural networks with LSTM based session encoders or classifiers having two LSTM hidden layers and adapt these baselines to our fraud detection task. Sel-CL performs a warm-up training by employing the SIMCLR contrastive loss. However, its augmentation strategy is image-specific. Hence, we employ the session reordering based augmentation strategy [3]. Both Sel-CL and CTRR perform label correction through sample similarity analysis. Since we are operating on sequential data, we perform session similarity analysis in the encoded representation space. Few-Shot and CLDet are insider threat detection approaches. Specifically, Few-Shot and CLDet employ BERT [54] and self-supervised contrastive learning models, respectively. DeepLog [16] and LogBert [48] are log anomaly detection approaches and employ LSTM and BERT models, respectively. All these four baselines (Few-Shot, CLDet, DeepLog, and LogBert) have not been originally designed for the noisy label learning task. We employ the same training set used for our CLFD framework to train all baselines.

B. Experimental Results

1) Overall Comparison: The overall comparison results for uniform and class dependent noise rates are shown in Tables I and II, respectively. Our CLFD framework noticeably outperforms against baselines w.r.t most of the performance metrics and specifically, at higher noise rates, we can observe that CLFD provides a significant performance improvement over baselines. Our CLFD framework addresses the dataset imbalance and session diversity challenges through supervised contrastive learning. Note that the performance of models trained under supervised contrastive loss can degrade under the noisy label setting. CLFD addresses this challenge by employing an effective label corrector which is specifically designed for the fraud detection task. Furthermore, CLFD effectively addresses the label correction uncertainty challenge through our proposed weighted supervised contrastive loss. DivMix and ULC are specifically designed for image datasets.

TABLE I: Performances of our CLFD and baselines (mean±std) for the uniform noise rate η. The higher the better for F1 and AUC-ROC. The lower the better for FPR. The best values for each noise rate are bold highlighted. DivMix and ULC are co-teaching based noise robust approaches, Sel-CL and CTRR are supervised contrastive learning based noise robust approaches, Few-Shot and CLDet are insider threat detection approaches, and DeepLog and LogBert are log anomaly detection approaches.

Models	η	**CERT**			**UMD-Wikipedia**			**Open-Stack**		
		F1	FPR	AUC-ROC	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC
DivMix	0.1	37.74±8.6	9.1±4.7	85.72±0.4	51.78±0.5	25.81±2.4	64.26±2.9	42.87±3.3	4.69±0.7	64.58±1.2
	0.2	22.71±0.3	20.66±3.5	84.07±0.9	28.58±2.9	21.93±0.9	53.85±1.6	39.11±2.5	5.24±1.5	61.89±1.2
	0.3	20.44±1.2	26.36±1.8	82.75±4.3	17.55±4.1	**2.69±1.2**	52.59±1.5	8.37±1.7	7.86±2.3	55.77±3.6
	0.45	14.04±3.6	37.32±7.8	74.48±5.7	10.19±1.8	6.54±2.7	50.72±1.9	6.63±1.6	5.54±1.2	50.21±0.6
ULC	0.1	53.35±4.6	11.15±1.5	84.78±2.7	53.60±1.1	18.58±1.4	65.88±1.8	41.12±2.4	7.26±1.7	64.95±3.6
	0.2	38.02±1.5	27.25±1.8	83.59±0.9	29.44±3.8	19.34±1.7	52.40±1.2	36.44±2.3	7.89±0.9	61.28±0.7
	0.3	24.14±8.4	19.32±7.8	80.62±3.1	23.17±2.2	19.25±2.1	50.41±1.2	10.87±2.4	6.26±2.7	53.96±1.3
	0.45	12.82±2.6	38.20±5.4	72.78±2.4	4.71±0.5	**4.08±0.4**	49.13±0.9	7.13±0.9	**5.13±1.2**	51.56±0.6
Sel-CL	0.1	73.96±1.8	5.15±1.8	82.62±0.8	70.93±3.4	14.11±2.8	77.28±1.9	48.82±0.9	15.15±0.8	63.91±1.9
	0.2	51.36±1.8	5.62±0.9	77.83±0.4	32.65±3.9	12.78±1.7	55.87±1.3	43.67±4.8	9.13±0.8	62.42±2.5
	0.3	46.17±1.3	11.20±0.8	76.95±0.9	26.72±1.5	16.59±2.1	52.08±1.4	39.31±1.6	13.80±2.9	58.83±1.4
	0.45	43.33±2.8	12.75±1.8	75.02±0.4	23.53±5.9	21.51±4.2	48.74±4.3	28.44±2.3	8.8±1.8	55.85±3.4
CTRR	0.1	69.72±4.1	7.25±1.7	82.88±0.4	66.95±1.7	14.66±0.8	75.88±0.4	31.48±3.7	8.3±0.8	63.84±0.7
	0.2	41.24±0.7	4.12±0.2	75.72±0.1	31.75±1.2	10.98±0.6	56.03±0.7	29.70±1.3	13.82±0.6	62.67±2.3
	0.3	24.61±2.5	13.65±2.1	74.03±1.4	23.93±1.9	16.76±0.6	51.73±1.1	22.33±5.9	15.02±1.2	58.57±4.3
	0.45	23.82±2.7	14.10±2.4	71.85±0.6	21.24±2.6	20.81±2.6	47.99±2.1	20.85±3.9	6.4±2.9	56.32±1.9
Few-Shot	0.1	37.29±8.5	44.88±1.3	59.13±6.8	43.82±1.4	45.93±0.6	52.48±1.7	9.56±1.3	4.82±0.5	52.82±0.8
	0.2	28.36±6.7	27.85±7.1	52.73±8.5	39.29±0.6	46.80±1.8	49.38±1.3	9.12±2.6	**2.26±0.4**	51.37±0.9
	0.3	21.93±5.9	32.14±1.3	47.91±4.3	37.16±0.6	50.24±3.1	48.69±1.5	20.78±1.9	7.81±1.7	50.36±0.6
	0.45	21.57±2.8	39.28±1.3	45.63±2.5	36.27±1.5	52.49±2.8	48.31±1.2	16.81±0.7	22.42±0.9	47.62±3.7
CLDet	0.1	67.72±4.1	2.14±1.1	82.78±0.3	37.53±0.9	8.69±0.7	60.87±1.3	56.07±0.9	4.85±2.4	78.96±1.5
	0.2	55.92±5.1	2.23±0.9	79.52±0.4	34.80±1.5	**2.25±0.6**	60.18±0.5	54.68±1.8	4.16±0.6	76.20±0.4
	0.3	30.65±2.9	6.42±1.4	71.86±0.6	27.74±1.8	3.46±1.4	57.29±0.7	48.96±0.6	**4.71±0.3**	73.26±0.8
	0.45	26.13±1.7	4.45±0.5	64.46±0.3	24.43±1.8	7.28±0.9	54.52±1.5	28.37±2.6	11.73±1.7	56.19±2.8
DeepLog	0.1	46.07±2.8	4.38±2.3	73.75±2.2	56.29±1.4	7.09±1.1	69.57±1.2	45.52±9.1	5.86±4.8	65.49±5.3
	0.2	33.35±1.5	4.61±1.2	66.23±1.9	37.28±1.8	4.48±1.5	62.17±1.4	29.78±7.2	4.35±3.9	58.97±5.7
	0.3	28.85±2.3	13.04±3.6	64.19±2.8	28.06±2.3	11.25±2.2	56.34±1.9	17.35±2.4	7.82±2.5	55.62±1.3
	0.45	16.72±1.4	12.32±2.2	58.64±2.1	13.06±2.9	8.29±1.8	51.71±3.6	10.74±3.8	5.71±3.2	50.68±2.9
LogBert	0.1	51.13±4.1	7.95±3.7	80.93±3.3	66.58±2.3	7.78±1.4	71.09±1.5	50.51±5.9	7.18±7.7	70.49±3.5
	0.2	35.81±2.8	7.49±2.1	69.56±1.7	50.72±3.7	14.63±5.6	61.49±3.4	35.82±5.6	13.37±4.4	56.47±3.7
	0.3	29.21±5.9	13.26±4.8	67.14±3.9	45.92±0.9	15.39±1.8	54.42±1.6	28.42±2.5	18.04±0.8	54.24±1.5
	0.45	22.47±2.2	9.11±1.4	65.09±2.1	33.67±1.9	12.46±1.3	49.44±0.7	15.58±2.8	15.67±4.4	50.67±3.7
CLFD	0.1	**77.93±4.3**	**1.32±0.2**	**90.72±0.3**	**75.17±0.5**	**5.83±0.9**	**80.79±0.6**	**64.54±1.8**	**4.52±2.4**	**88.96±2.1**
	0.2	**75.51±4.7**	**1.95±0.4**	**88.48±0.2**	**57.01±2.9**	3.81±0.5	**69.63±1.6**	**62.77±2.3**	5.62±1.7	**88.54±2.8**
	0.3	**70.67±3.6**	**2.13±0.2**	**87.61±0.3**	**55.57±2.7**	5.30±0.7	**68.74±1.5**	**59.72±1.2**	5.79±1.6	**86.78±1.2**
	0.45	**62.77±2.9**	**2.53±0.5**	**85.76±0.8**	**52.89±1.6**	5.52±0.6	**67.22±0.7**	**48.89±2.3**	5.46±0.7	**78.35±1.6**

TABLE II: Performances of our CLFD and baselines (mean±std) for the class dependent noise rates $\eta_{10} = 0.3$ and $\eta_{01} = 0.45$.

Models	**CERT**			**UMD-Wikipedia**			**Open-Stack**		
	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC
DivMix	17.22±1.3	30.11±2.7	75.06±0.3	5.95±0.7	6.83±3.7	48.73±1.8	8.77±2.1	5.35±0.5	51.23±0.7
ULC	21.33±2.8	27.49±1.7	72.26±3.9	12.01±0.4	**5.25±2.7**	51.57±1.4	5.23±1.6	**4.81 ±0.8**	49.12±1.3
Sel-CL	38.41±5.9	18.75±5.3	75.48±1.8	18.19±2.3	22.56±5.3	46.03±2.7	35.36±1.7	23.53±2.3	64.32±1.6
CTRR	23.35±3.4	16.35±3.2	75.96±1.2	19.84±1.3	23.14±0.7	46.57±0.9	32.15±1.8	22.95±3.3	59.84±0.7
Few-Shot	24.19±7.2	36.28±1.7	48.81±7.9	40.95±1.7	51.06±1.5	49.85±1.5	19.96±2.6	15.84±6.1	49.52±3.2
CLDet	27.43±1.6	9.34±1.9	59.83±1.2	21.53±2.7	9.27±0.7	54.03±0.8	29.39±3.7	10.59±2.5	54.12±1.4
DeepLog	25.86±2.4	10.27±1.5	64.81±1.6	21.37±3.5	14.04±1.7	55.69±4.6	16.10±1.8	5.03±1.4	52.94±0.8
LogBert	28.51±1.9	16.92±1.7	68.77±2.3	38.87±4.6	17.34±3.7	56.32±4.3	21.85±1.3	17.26±1.8	51.59±2.3
CLFD	**60.77±2.8**	**1.90±0.7**	**82.55±0.6**	**58.79±3.6**	6.50±1.7	**70.34±2.2**	**48.45±3.4**	6.65±2.2	**76.35±1.1**

TABLE III: Performances of our label corrector on the noisy training set $\widetilde{\mathcal{T}}$ (mean±std). TPR and TNR denote True Positive Rate and True Negative Rate, respectively. Higher the better for TPR and TNR.

Dataset	$\eta = 0.45$		$\eta_{10} = 0.3$ and $\eta_{01} = 0.45$	
	TPR	TNR	TPR	TNR
CERT	70.25±2.3	90.69±1.7	79.42±1.6	87.47±1.4
UMD-Wikipedia	71.73±0.7	89.38±1.3	79.61±1.7	88.34±2.1
Open-Stack	72.62±1.5	93.22±2.4	80.52±3.6	88.46±2.8

Hence, at higher noise rates, due to the session diversity challenge, their performances degrade significantly. Sel-CL and CTRR employ supervised contrastive learning models. Hence, they are expected to effectively address dataset imbalance and session diversity challenges. However, they perform label correction through sample similarity analysis. At higher noise rates, due to the session diversity challenge, corrected labels of many sessions do not match the ground truth [7]. Furthermore, they do not specifically address the label correction uncertainty challenge. As a result of these improper learning effects, both Sel-CL and CTRR underperform. The remaining baselines CLDet, Few-Shot, DeepLog, and LogBert also show poor performances at higher noise rates. These remaining baselines do not employ effective noise robust mechanisms in their design. Therefore, they are sensitive to the noisy label setting.

2) Label Corrector Performance Analysis: We analyze the performance of our label corrector on the noisy training set $\widetilde{\mathcal{T}}$. We compare the predictions of the label corrector with the corresponding ground truth labels. This empirical analysis result is shown in the Table III. Clearly, our label corrector substantially reduces the original dataset noise, and provides better quality supervision to the fraud detector when compared to the original noisy labels.

3) Training Latency Analysis: All experiments are executed on GPU Tesla V100 (32GB RAM) and CPU Xeon 6258R 2.7 GHz with 226 GB hard disk. The training latencies (in seconds) for our CLFD framework are 30,816 (CERT), 19,158 (UMD-Wikipedia), and 28,872 (Open-Stack). Both Sel-CL and CTRR baselines also incur similar training latencies due to the employment of supervised contrastive learning models. However, CLFD incurs around 4 times more training latency than the remaining baselines. This is because the remaining baselines do not employ supervised contrastive learning models. Even though the supervised contrastive learning model incurs higher training costs, it can effectively address session diversity and dataset imbalance challenges in the fraud detection task.

4) Ablation Analysis: We conduct the ablation analysis study on our CLFD framework by ablating the following main components: Label Corrector (LC), $l^{\lambda}_{GCE}(\cdot,\cdot)$, GCE loss, Fraud Detector (FD), $\mathcal{L}_{Sup}$, and classifier in the FD. The ablation analysis results for uniform and class dependent noise rates are shown in Tables IV and V, respectively.

W/o LC. We directly train the fraud detector on noisy labels by using the vanilla supervised contrastive loss [5]. Then, we train the classifier in the fraud detector by employing the noisy labels. For the uniform noise rate, the mean F1 scores drop to 25.53 (CERT), 23.29 (UMD-Wikipedia), and 38.35 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 16.46 (CERT), 32.69 (UMD-Wikipedia), and 36.16 (Open-Stack). The performance of the employed supervised contrastive learning model in the fraud detector can significantly degrade under high noise rates [7]. The reason is that many sessions that belong to different classes in the ground truth are pushed closer in the encoded representation space. Due to this improper learning effect, the performance degrades significantly.

W/o $l^{\lambda}_{GCE}(\cdot,\cdot)$. We employ the vanilla GCE loss $l_{GCE}(\cdot,\cdot)$ shown in Equation 1 instead of our proposed $l^{\lambda}_{GCE}(\cdot,\cdot)$ for training the classifiers in both label corrector and fraud detector. For the uniform noise rate, the mean F1 scores drop to 53.44 (CERT), 46.83 (UMD-Wikipedia), and 41.53 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 46.46 (CERT), 52.78 (UMD-Wikipedia), and 44.74 (Open-Stack). The vanilla GCE loss does not specifically address the label memorization issue. Applying mixup based data augmentation can aid in alleviating the label memorization issue [51]. Hence, mixup version of the GCE loss provides better noise robustness when compared to the vanilla GCE loss.

W/o GCE. We employ the cross entropy loss instead of our proposed $l^{\lambda}_{GCE}(\cdot,\cdot)$ for training the classifiers in both label corrector and fraud detector. For the uniform noise rate, the mean F1 scores drop to 7.35 (CERT), 19.40 (UMD-Wikipedia), and 9.28 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 15.21 (CERT), 17.18 (UMD-Wikipedia), and 10.48 (Open-Stack). Clearly, we can observe a significant drop in the performance scores. Cross entropy loss is typically sensitive to the label noise which can result in model over-fitting [13]. Additionally, it is an unbounded loss which exacerbates the model over-fitting issue.

W/o FD. We directly deploy our trained label corrector for the test case inference. For the uniform noise rate, the mean F1 scores drop to 42.78 (CERT), 36.98 (UMD-Wikipedia), and 38.55 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 40.77 (CERT), 47.87 (UMD-Wikipedia), and 39.73 (Open-Stack). The label corrector employs self-supervised pre-training component in which, the augmented sessions generated from a given session are brought closer in the representation space. This component can effectively address the dataset imbalance challenge [3]. However, to effectively address the session diversity challenge, we require the supervised contrastive learning approach.

W/o $\mathcal{L}_{Sup}$. We employ the unweighted version of $\mathcal{L}_{Sup}$ which is denoted as $\mathcal{L}^{uw}_{Sup}$ (refer to Section VII) to train our session encoder. For the uniform noise rate, the mean F1 scores drop to 48.73 (CERT), 44.31 (UMD-Wikipedia), and 45.01 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 44.69 (CERT), 50.56 (UMD-Wikipedia), and 43.47 (Open-Stack). $\mathcal{L}^{uw}_{Sup}$ does not specifically address the label correction uncertainty challenge. As a consequence,

TABLE IV: Ablation analysis results (mean±std) for the uniform noise rate $\eta = 0.45$.

Models	CERT			UMD-Wikipedia			Open-Stack		
	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC
CLFD	62.77±2.9	2.53±0.5	85.76±0.8	52.89±1.6	5.52±0.6	67.22±0.7	48.89±2.3	5.46±0.7	78.35±1.6
w/o LC	25.53±2.4	9.42±2.5	71.57±0.8	23.29±1.3	8.84±1.7	53.35±0.9	38.35±0.9	4.68±0.4	65.43±1.3
w/o $l^{\lambda}_{GCE}(\cdot,\cdot)$	53.44±2.3	3.90±0.4	81.93±1.2	46.83±1.8	6.79±1.4	62.52±1.6	41.53±4.4	14.11±2.7	71.97±1.4
w/o GCE loss	7.35±1.1	7.71±0.8	52.19±1.1	19.40±1.6	10.04±0.3	52.44±0.9	9.28±1.5	10.98±1.1	51.06±1.9
w/o FD	42.78±5.5	7.37±1.8	78.48±1.9	36.98±1.7	7.66±0.7	61.62±1.9	38.55±1.6	5.96±1.9	62.82±1.5
w/o $\mathcal{L}_{Sup}$	48.73±1.8	5.12±0.5	81.08±0.9	44.31±1.6	6.64±0.7	62.89±0.5	45.01±2.1	5.62±1.2	66.43±1.9
w/o classifier (FD)	46.65±2.9	3.24±1.8	79.67±1.3	43.89±2.4	6.54±1.5	62.81±1.3	41.13±2.8	4.59±1.6	63.70±1.7

TABLE V: Ablation analysis results (mean±std) for the class dependent noise rates $\eta_{10} = 0.3$ and $\eta_{01} = 0.45$.

Models	CERT			UMD-Wikipedia			Open-Stack		
	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC	F1	FPR	AUC-ROC
CLFD	60.77±2.8	1.90±0.7	82.55±0.6	58.79±3.6	6.50±1.7	70.34±2.2	48.45±3.4	6.65±2.2	76.35±1.1
w/o LC	16.46±6.5	15.96±2.8	59.18±3.2	32.69±4.4	15.35±2.3	56.10±1.8	36.16±4.8	11.62±1.7	57.62±1.5
w/o $l^{\lambda}_{GCE}(\cdot,\cdot)$	46.46±3.4	6.61±2.9	79.86±2.8	52.78±2.3	8.46±1.9	67.69±1.6	44.74±3.5	9.23±1.4	71.77±1.5
w/o GCE loss	15.21±4.3	8.63±1.8	59.58±2.6	17.18±1.9	8.52±2.7	52.29±2.1	10.48±3.2	9.45±2.3	52.55±2.6
w/o FD	40.77±2.6	5.14±1.4	74.03±1.6	47.87±2.9	5.89±2.7	62.17±1.5	39.73±1.5	5.34±0.7	66.38±0.9
w/o $\mathcal{L}_{Sup}$	44.69±3.8	6.43±2.8	78.83±1.4	50.56±2.3	7.62±3.1	64.67±2.4	43.47±4.9	9.30±3.4	70.65±2.9
w/o classifier (FD)	43.13±1.7	8.06±2.4	77.96±1.7	48.12±1.4	12.26±2.9	63.76±2.3	42.25±1.9	7.88±0.9	69.33±1.2

there is a noticeable drop in the performance.

W/o classifier (FD). We estimate the center (mean) of label corrected normal and malicious sessions in the encoded representation space corresponding to the session encoder. A test session is classified based on its proximities to both these centers [4]. For the uniform noise rate, the mean F1 scores drop to 46.65 (CERT), 43.89 (UMD-Wikipedia), and 41.13 (Open-Stack). Similarly, for the class dependent noise rates, the mean F1 scores drop to 43.13 (CERT), 48.12 (UMD-Wikipedia), and 42.25 (Open-Stack). The cluster centers are estimated by using the corrected labels. At high noise rates, due to the label correction uncertainty challenge, these estimated cluster centers can noticeably deviate from their ground-truth counterparts. Hence, we can observe a noticeable drop in the performance.

V. Conclusion

In this work, we have developed a supervised contrastive learning based fraud detection framework called CLFD, which operates in the noisy label setting. We proposed a mixup version of the GCE loss to train classifiers in our label corrector and fraud detector, and theoretically showed its efficacy. We proposed a weighted supervised contrastive loss to train the session encoder in our fraud detector and theoretically showed that our proposed loss is upper bounded by the ideal loss. We developed a training procedure to train our fraud detector to learn separable session representations for the noisy label learning task. The empirical study on three benchmark datasets demonstrated that our CLFD can outperform state-of-the-art baselines. In our future work, we plan to extend CLFD to model session specific noise rates. We will explore benefits of developing the mixup versions of other robust loss functions. We will also explore benefits of integrating supervised contrastive learning model with co-teaching based noisy label learning approaches.

Acknowledgement

This work was supported in part by NSF grants 1920920, 1946391, and 2103829.

VI. Proofs of Theoretical Results

A. Proof for Theorem 1

We can express $l^{\lambda}_{GCE}(\cdot,\cdot)$ as:

$$\lim_{q\to 0} l^{\lambda}_{GCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right) = \lim_{q\to 0}\left[\sum_{k=0}^{1}\frac{\widetilde{m}_{ik}}{q}\left(1 - f_k(\mathbf{v}_i^{\lambda})^q\right)\right] = \sum_{k=0}^{1}\widetilde{m}_{ik}\lim_{q\to 0}\left[\frac{1 - f_k(\mathbf{v}_i^{\lambda})^q}{q}\right] \quad (10)$$

By using the result from [13], which states that for any encoded session representation $\mathbf{v}$, we have that: $\lim_{q\to 0}\left[\frac{1-f_k(\mathbf{v})^q}{q}\right] = -\log\left(f_k(\mathbf{v})\right)$. We can rewrite Equation 10 as:

$$\lim_{q\to 0} l^{\lambda}_{GCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right) = -\sum_{k=0}^{1}\widetilde{m}_{ik}\log\left(f_k(\mathbf{v}_i^{\lambda})\right) = l^{\lambda}_{CCE}\left(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i\right)$$

B. Proof for Theorem 2

First we will derive the upper bound. We use the result from [13] which states that for any encoded session representation $\mathbf{v}$: $\sum_{k=0}^{1}\frac{1-f_k(\mathbf{v})^q}{q} \leq \frac{1}{q}$. Since $\widetilde{m}_{ik} \in [0,1]$, we can derive the upper bound as:

$$l^{\lambda}_{GCE}(f(\mathbf{v}_i^{\lambda}), \widetilde{\mathbf{m}}_i) = \sum_{k=0}^{1}\frac{\widetilde{m}_{ik}}{q}\left(1 - f_k(\mathbf{v}_i^{\lambda})^q\right) \leq \frac{1}{q}$$

Now we will derive the lower bound. From our mixup strategy outlined in Section III-A1, we can infer the result:

$$\widetilde{m}_{ik} = \begin{cases} \lambda & \text{if} \quad \widetilde{e}_{ik} = 1 \\ 1-\lambda & \text{otherwise} \end{cases}$$

Thus, we can derive the result:

$$l^{\lambda}_{GCE}(f(\mathbf{v}^{\lambda}_i), \widetilde{\mathbf{m}}_i) = \sum_{k=0}^{1} \frac{\widetilde{m}_{ik}}{q}\left(1 - f_k(\mathbf{v}^{\lambda}_i)^q\right) \geq \min(\lambda, 1-\lambda) \sum_{k=0}^{1} \frac{1 - f_k(\mathbf{v}^{\lambda}_i)^q}{q} \quad (11)$$

We use the result from [13] which states that for any encoded session representation $\mathbf{v}$: $\sum_{k=0}^{1} \frac{1-f_k(\mathbf{v})^q}{q} \geq \frac{2-2^{(1-q)}}{q}$. Thus, we can rewrite Equation 11 as:

$$l^{\lambda}_{GCE}(f(\mathbf{v}^{\lambda}_i), \widetilde{\mathbf{m}}_i) \geq \min(\lambda, 1-\lambda)\frac{2-2^{(1-q)}}{q}$$

C. Proof for Theorem 3

By applying expectation conditioning on the noisy risk $\widetilde{R}^{\lambda}_{GCE}(f)$, we can derive the result:

$$\begin{aligned} \widetilde{R}^{\lambda}_{GCE}(f) &= \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right)\right] \\ &= \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i = \mathbf{m}_i \right] P(\widetilde{\mathbf{m}}_i = \mathbf{m}_i) \\ &\quad + \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i \neq \mathbf{m}_i \right] P(\widetilde{\mathbf{m}}_i \neq \mathbf{m}_i) \\ &= R^{\lambda}_{GCE}(f)(1-\eta) + \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i \neq \mathbf{m}_i \right] \eta \end{aligned} \quad (12)$$

Since the bound shown in Theorem 2 holds for both cases when $\widetilde{\mathbf{m}}_i = \mathbf{m}_i$ or $\widetilde{\mathbf{m}}_i \neq \mathbf{m}_i$, by plugging the upper bound in Theorem 2 inside Equation 12, we get the result:

$$\begin{aligned} \widetilde{R}^{\lambda}_{GCE}(f) &\leq R^{\lambda}_{GCE}(f)(1-\eta) + \frac{\eta}{q} \\ &\leq R^{\lambda}_{GCE}(f) + \frac{\eta}{q} \end{aligned}$$

D. Proof for Theorem 4

By applying expectation conditioning on the noisy risk $\widetilde{R}^{\lambda}_{GCE}(f)$, we can derive the result:

$$\begin{aligned} \widetilde{R}^{\lambda}_{GCE}(f) &= \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right)\right] \\ &= \sum_{\widetilde{y}=0}^{1} \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i = \mathbf{m}_i, \widetilde{y}_i = \widetilde{y} \right] P(\widetilde{\mathbf{m}}_i = \mathbf{m}_i, \widetilde{y}_i = \widetilde{y}) \\ &\quad + \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i \neq \mathbf{m}_i, \widetilde{y}_i = \widetilde{y} \right] P(\widetilde{\mathbf{m}}_i \neq \mathbf{m}_i, \widetilde{y}_i = \widetilde{y}) \\ &= R^{\lambda}_{GCE}(f|y_i = 1)(1-\eta_{10})\widetilde{\tau}^1 + \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i \neq \mathbf{m}_i, \widetilde{y}_i = 1 \right] \eta_{10}\widetilde{\tau}^1 \\ &\quad + R^{\lambda}_{GCE}(f|y_i = 0)(1-\eta_{01})\widetilde{\tau}^0 + \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[l^{\lambda}_{GCE}\left(f\left(\mathbf{v}^{\lambda}_i\right), \widetilde{\mathbf{m}}_i\right) \middle| \widetilde{\mathbf{m}}_i \neq \mathbf{m}_i, \widetilde{y}_i = 0 \right] \eta_{01}\widetilde{\tau}^0 \end{aligned} \quad (13)$$

By using Theorem 2, we can rewrite Equation 13 as:

$$\begin{aligned} \widetilde{R}^{\lambda}_{GCE}(f) &\leq R^{\lambda}_{GCE}(f|y_i = 1)(1-\eta_{10})\widetilde{\tau}^1 + \frac{\eta_{10}\widetilde{\tau}^1}{q} \\ &\quad + R^{\lambda}_{GCE}(f|y_i = 0)(1-\eta_{01})\widetilde{\tau}^0 + \frac{\eta_{01}\widetilde{\tau}^0}{q} \\ &\leq \widetilde{\tau}^1\left(R^{\lambda}_{GCE}(f|y_i = 1) + \frac{\eta_{10}}{q}\right) + \widetilde{\tau}^0\left(R^{\lambda}_{GCE}(f|y_i = 0) + \frac{\eta_{01}}{q}\right) \end{aligned}$$

E. Proof for Theorem 5

By applying expectation conditioning on $\mathcal{L}_{Sup}$ using the label corrector confidence, we can rewrite Equation 9 as:

$$\begin{aligned} &\mathcal{L}_{Sup} \\ &= \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[\frac{1}{|B(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in B(\mathbf{x}_i)} (c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \middle| c_i \approx 1 \right] P(c_i \approx 1) \\ &+ \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[\frac{1}{|B(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in B(\mathbf{x}_i)} (c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \middle| c_i \not\approx 1 \right] P(c_i \not\approx 1) \end{aligned} \quad (14)$$

Here, $c_i \not\approx 1$ denotes that c_i is not closer to 1. Let $B^h(\mathbf{x}_i) = \{\mathbf{x}_p \in B(\mathbf{x}_i) | c_p \approx 1\}$ and $B^l(\mathbf{x}_i) = \{\mathbf{x}_p \in B(\mathbf{x}_i) | c_p \not\approx 1\}$. We can rewrite the first term in the right hand side of Equation 14 as:

$$\begin{aligned} &\mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \left[\frac{1}{|B(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in B(\mathbf{x}_i)} (c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \middle| c_i \approx 1 \right] P(c_i \approx 1) \\ &= \frac{1}{|B(\mathbf{x}_i)|} \mathop{\mathbb{E}}_{\mathbf{x}_i \sim \mathcal{D}} \Bigg[\sum_{\substack{\mathbf{x}_p \in B(\mathbf{x}_i) \\ c_p \approx 1}} (c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \\ &\qquad + \sum_{\substack{\mathbf{x}_r \in B(\mathbf{x}_i) \\ c_r \not\approx 1}} (c_i c_r)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_r\right) \middle| c_i \approx 1 \Bigg] P(c_i \approx 1) \\ &= \frac{1}{|B(\mathbf{x}_i)|} \Bigg\{ |B^h(\mathbf{x}_i)| \mathcal{L}_{Orc} \\ &\qquad + |B^l(\mathbf{x}_i)| \mathop{\mathbb{E}}_{\substack{\mathbf{x}_i \sim \mathcal{D} | c_i \approx 1 \\ c_p \not\approx 1}} \left[(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \Bigg\} P(c_i \approx 1) \\ &= \Bigg\{ P(c \approx 1)\mathcal{L}_{Orc} \\ &\qquad + P(c \not\approx 1) \mathop{\mathbb{E}}_{\substack{\mathbf{x}_i \sim \mathcal{D} | c_i \approx 1 \\ c_p \not\approx 1}} \left[(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \Bigg\} P(c_i \approx 1) \\ &\leq P(c \approx 1) \Bigg\{ P(c \approx 1)\mathcal{L}_{Orc} + \mathop{\mathbb{E}}_{\substack{\mathbf{x}_i \sim \mathcal{D} | c_i \approx 1 \\ c_p \not\approx 1}} \left[(c_i c_p)\, l_{Sup}\left(\mathbf{z}_i, \mathbf{z}_p\right) \right] \Bigg\} \end{aligned} \quad (15)$$

Similarly, we can rewrite the second term in the right hand side of Equation 14 as:

$$\mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}}\left[\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}(c_ic_p)\,l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)\middle|c_i\not\approx 1\right]P(c_i\not\approx 1)$$
$$=\mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}|c_i\not\approx 1}\left[(c_ic_p)\,l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)\right]P(c_i\not\approx 1)$$
$$\leq\mathop{\mathbb{E}}_{\mathbf{x}_i\sim\mathcal{D}|c_i\not\approx 1}\left[(c_ic_p)\,l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)\right] \quad (16)$$

By plugging the results shown in Equations 15 and 16 in the right hand side of Equation 14, the theorem follows.

VII. Theoretical Underpinnings on $\mathcal{L}_{Sup}$

For the session encoder parameter set $\boldsymbol{\theta}$, the gradient of $\mathcal{L}_{Sup}$ w.r.t to $\theta_j\in\boldsymbol{\theta}$ is given by:

$$\frac{\partial\mathcal{L}_{Sup}}{\partial\theta_j}=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}(c_ic_p)\,\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j} \quad (17)$$

A possible variant is the unweighted version of $\mathcal{L}_{Sup}$. This loss and its corresponding gradient is given by:

$$\mathcal{L}_{Sup}^{uw}=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right) \quad (18)$$

$$\frac{\partial\mathcal{L}_{Sup}^{uw}}{\partial\theta_j}=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j} \quad (19)$$

Consider a scenario where the label corrector provides highly confident predictions ($c\approx 1$) for most of the training set sessions. By comparing Equations 17 and 19, we have the result: $\frac{\partial\mathcal{L}_{Sup}^{uw}}{\partial\theta_j}\approx\frac{\partial\mathcal{L}_{Sup}}{\partial\theta_j}$. This result states that in this scenario, after the training loop, both losses $\mathcal{L}_{Sup}^{uw}$ and $\mathcal{L}_{Sup}$ can produce similar estimations for the parameter set $\boldsymbol{\theta}$. However, consider another scenario where the label corrector does not provide highly confident predictions for most of the training set sessions. In this scenario, a large amount of session pairs which are predicted to belong to the same class, actually belong to different classes in the ground truth. From Equation 19, it is clear that if we apply $\mathcal{L}_{Sup}^{uw}$ for session representation learning then, many sessions which belong to different classes in the ground truth are pushed closer in the encoded representation space, which leads to improper learning effect. However, from Equation 17, we can infer that this improper learning effect is reduced when we apply $\mathcal{L}_{Sup}$ due to the usage of session pair weight c_ic_p.

Another attractive variant is to discard/filter those session pairs $(\mathbf{x}_i,\mathbf{x}_p)$ from the training loop which are predicted to belong to the same class with a confidence falling below the given threshold τ. To achieve this goal, we can design another supervised contrastive loss which is given by:

$$\mathcal{L}_{Sup}^{ftr}=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}\mathbb{I}\left(c_ic_p>\tau\right)l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right) \quad (20)$$

Here, $\mathbb{I}(\cdot)$ denotes the indicator function. The gradient for this loss is given by:

$$\frac{\partial\mathcal{L}_{Sup}^{ftr}}{\partial\theta_j}=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}\mathbb{I}\left(c_ic_p>\tau\right)\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j} \quad (21)$$

From Equation 21, we can infer that if $(\mathbf{x}_i,\mathbf{x}_p)$ are predicted to belong to the same class with low confidence $(c_ic_p\leq\tau)$ then, the gradient value contributed by this pair which is given by: $\frac{\partial l_{Sup}(\mathbf{z}_i,\mathbf{z}_p)}{\partial\theta_j}$, is discarded. We analyze the gradient of $\mathcal{L}_{Sup}^{ftr}$ by taking expectation w.r.t $\mathbb{I}\left(c_ic_p>\tau\right)$, which is given by:

$$\mathop{\mathbb{E}}_{\mathbb{I}(c_ic_p>\tau)}\left[\frac{\partial\mathcal{L}_{Sup}^{ftr}}{\partial\theta_j}\right]$$
$$=\mathop{\mathbb{E}}_{\mathbb{I}(c_ic_p>\tau)}\left[\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}\mathbb{I}\left(c_ic_p>\tau\right)\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j}\right]$$
$$=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}\mathop{\mathbb{E}}_{\mathbb{I}(c_ic_p>\tau)}\left[\mathbb{I}\left(c_ic_p>\tau\right)\right]\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j}$$
$$=\frac{1}{R}\sum_{i=1}^{R}\frac{1}{|B(\mathbf{x}_i)|}\sum_{\mathbf{x}_p\in B(\mathbf{x}_i)}P\left(c_ic_p>\tau\right)\frac{\partial l_{Sup}\left(\mathbf{z}_i,\mathbf{z}_p\right)}{\partial\theta_j} \quad (22)$$

By using this result and by comparing Equations 22 and 17, we can make the following observations: In the scenario where the label corrector does not provide highly confident predictions for most of the training set sessions, we have that: $\mathop{\mathbb{E}}_{\mathbb{I}(c_ic_p>\tau)}\left[\frac{\partial\mathcal{L}_{Sup}^{ftr}}{\partial\theta_j}\right]\approx\frac{\partial\mathcal{L}_{Sup}}{\partial\theta_j}$. This approximation also holds when the label corrector provides highly confident predictions for most of the training set sessions. This result states that in both these scenarios, after the training loop, $\mathcal{L}_{Sup}^{ftr}$ stochastically produces similar estimations as $\mathcal{L}_{Sup}$ for the parameter set $\boldsymbol{\theta}$. The performance of $\mathcal{L}_{Sup}^{ftr}$ is influenced by the setting of the hyper-parameter τ. Setting high values ($\tau\approx 1$) can result in filtering out most of the session pairs. As a consequence, the session diversity challenge cannot be effectively addressed which can lead to poor decision boundaries. Setting low values ($\tau\approx 0.5$) can result in pushing many sessions belonging to opposite classes in the ground truth, closer in the encoded representation space. Since the training set is imbalanced and large, it becomes computationally challenging to identify optimal setting for τ. However, $\mathcal{L}_{Sup}$ elegantly overcomes these challenges by employing c_ic_p as weights to modulate the learning effect from low/high confident session pairs.

REFERENCES

[1] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Computers & Security*, 2021.

[2] S. Yuan, P. Zheng, X. Wu, and H. Tong, "Few-shot insider threat detection," in *The 29th ACM International Conference on Information and Knowledge Management*, 2020.

[3] M. S. Vinay, S. Yuan, and X. Wu, "Contrastive learning for insider threat detection," in *Database Systems for Advanced Applications - 27th International Conference*, 2022.

[4] M. Vinay, S. Yuan, and X. Wu, "Fraud detection via contrastive positive unlabeled learning," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022.

[5] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Conference on Neural Information Processing Systems*, 2020.

[6] Y. Zhao, G. Zheng, S. Mukherjee, R. McCann, and A. H. Awadallah, "Admoe: Anomaly detection with mixture-of-experts from noisy labels," *CoRR*, vol. abs/2208.11290, 2022.

[7] Z. Huang, J. Zhang, and H. Shan, "Twin contrastive learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] S. Li, X. Xia, S. Ge, and T. Liu, "Selective-supervised contrastive learning with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] L. Yi, S. Liu, Q. She, A. I. McLeod, and B. Wang, "On learning contrastive representations for learning with noisy labels," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.

[10] Y. Huang, B. Bai, S. Zhao, K. Bai, and F. Wang, "Uncertainty-aware learning against label noise on imbalanced datasets," in *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, 2022.

[11] E. Zheltonozhskii, C. Baskin, A. Mendelson, A. M. Bronstein, and O. Litany, "Contrast to divide: Self-supervised pre-training for learning with noisy labels," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[12] J. Li, C. Xiong, and S. C. Hoi, "Learning from noisy data with robust representation learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[13] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Conference on Neural Information Processing Systems*, 2018.

[14] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," in *IEEE Symposium on Security and Privacy Workshops*, 2013.

[15] S. Kumar, F. Spezzano, and V. Subrahmanian, "Vews: A wikipedia vandal early warning system," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[16] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS*, 2017.

[17] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[18] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[19] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[20] X. Wang, E. Kodirov, Y. Hua, and N. M. Robertson, "Improving MAE against CCE under label noise," *CoRR*, vol. abs/1903.12141, 2019.

[21] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[22] Y. Xu, P. Cao, Y. Kong, and Y. Wang, "L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise," in *Neural Information Processing Systems*, 2019.

[23] D. Ortego, E. Arazo, P. Albert, N. E. O'Connor, and K. McGuinness, "Multi-objective interpolation training for robustness to label noise," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in *International Conference on Machine Learning*, 2019.

[25] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *International Conference on Learning Representations*, 2017.

[26] G. Patrini, A. Rozza, A. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, "Are anchor points really indispensable in label-noise learning?" in *Advances in Neural Information Processing Systems*, 2019.

[29] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems*, 2018.

[30] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*, 2018.

[31] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *International Conference on Learning Representations*, 2020.

[32] T. Kim, J. Ko, S. Cho, J. Choi, and S.-Y. Yun, "FINE samples for learning with noisy labels," in *Advances in Neural Information Processing Systems*, 2021.

[33] J. Li, C. Xiong, and S. C. Hoi, "Mopro: Webly supervised learning with momentum prototypes," in *International Conference on Learning Representations*, 2021.

[34] S. Liu, Z. Zhu, Q. Qu, and C. You, "Robust training under label noise by over-parameterization," in *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[35] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *International Conference on Learning Representations*, 2015.

[36] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[38] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *CoRR*, vol. abs/2011.00362, 2020.

[39] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *Computational Science - 18th International Conference*, 2018.

[40] L. Lin, S. Zhong, C. Jia, and K. Chen, "Insider threat detection based on deep belief network feature representation," in *International Conference on Green Informatics*, 2017.

[41] J. Lu and R. K. Wong, "Insider threat detection with long short-term memory," in *Proceedings of the Australasian Computer Science Week Multi-conference*, 2019.

[42] D. Zhang, Y. Zheng, Y. Wen, Y. Xu, J. Wang, Y. Yu, and D. Meng, "Role-based log analysis applying deep learning for insider threat detection," in *Proceedings of the 1st Workshop on Security-Oriented Designs of Computer Architectures and Processors*, 2018.

[43] J. Jiang, J. Chen, T. Gu, K.-K. R. Choo, C. Liu, M. Yu, W. Huang, and P. Mohapatra, "Anomaly detection with graph convolutional networks for insider threat and fraud detection," in *MILCOM IEEE Military Communications Conference (MILCOM)*, 2019.

[44] T. Rashid, I. Agrafiotis, and J. R. Nurse, "A new take on detecting insider threats: Exploring the use of hidden markov models," in *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, 2016.

[45] S. Yuan, P. Zheng, X. Wu, and Q. Li, "Insider threat detection via hierarchical neural temporal point processes," in *IEEE International Conference on Big Data (Big Data)*, 2019.

[46] S. Zhou, L. Wang, J. Yang, and P. Zhan, "Sitd: Insider threat detection using siamese architecture on imbalanced data," in *IEEE 25th International Conference on Computer Supported Cooperative Work in Design*, 2022.
[47] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Machine Learning with Applications*, 2023.
[48] H. Guo, S. Yuan, and X. Wu, "Logbert: Log anomaly detection via bert," in *International Joint Conference on Neural Networks (IJCNN)*, 2021.
[49] P. Jia, S. Cai, B. C. Ooi, P. Wang, and Y. Xiong, "Robust and transferable log-based anomaly detection," *Proc. ACM Manag. Data*, 2023.
[50] Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, and H. Ma, "CLEAR: contrastive learning for sentence representation," *CoRR*, vol. abs/2012.15466, 2020.
[51] H. Cheng, Z. Zhu, X. Sun, and Y. Liu, "Mitigating memorization of noisy labels via regularization between representations," in *The Eleventh International Conference on Learning Representations*, 2023.
[52] G. Blanchard, M. Flaska, G. Handy, S. Pozzi, and C. Scott, "Classification with asymmetric label noise: Consistency and maximal denoising," *Electronic Journal of Statistics*, 2016.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.
[54] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.