Crossing Domains with the Inductive Transfer Encoder: Case Study in Keystroke Biometrics

John V. Monaco and Manuel M. Vindiola
U.S. Army Research Laboratory
Aberdeen, MD
{john.v.monaco2.ctr,manuel.m.vindiola.civ}@mail.mil

Abstract

Keystroke biometric samples are often collected under various conditions, such as different device types, increasing levels of practice through repetition, and subject impairment. Cross-domain comparisons, in which query samples are collected under different conditions than the template, generally lead to degraded performance. The difficulty in comparing samples from different domains can be viewed as an inductive transfer learning problem, in which general knowledge of the mapping between a source and target domain can be applied to increase task performance, such as verification accuracy, by transferring source domain samples to the target domain. In this light, we propose the inductive transfer encoder, which utilizes pairwise correspondences from an independent dataset to learn a general transformation between domains. When the transformation is applied to template samples in the source domain, and query samples are in the target domain, increased verification performance is observed. We evaluate four different strategies for establishing the pairwise correspondences between source and target domains, and two cross-domain problems: low vs high practice levels and one vs two typing hands. Empirical results demonstrate that the inductive transfer encoder captures general rules that can be applied to transfer source domain samples to the target domain.

1. Introduction

The performance of a keystroke biometric verification system depends a number of factors, including the detection algorithm, conditions under which samples are obtained, and subject-to-subject differences [11]. Degradation of verification accuracy can be linked to cross-domain sample comparisons, i.e, the subject's template was collected under different conditions than the genuine query samples. This situation arises when subjects are enrolled on a different device than is used for testing [15], switch between free-text and fixed-text input tasks [15], or due to impairments such as only being able to type with one hand [13].

It is also commonly understood that a subject's typing behavior changes over time, which leads to degraded performance when comparing genuine query samples to a template collected earlier [6]. In this scenario, the domain of the query and template samples may correspond to high and low levels of practice, respectively. In an adaptive biometric system, template update strategies can be used to improve asymptotic accuracy as the subject's template is updated over time to reflect the change in typing behavior [6].

The challenge of comparing biometric samples from different domains can be cast as a transfer learning problem. Transfer learning attempts to improve the performance of a target task, such as verification accuracy, by utilizing external knowledge [14]. Inductive transfer learning is a subfield of transfer learning, in which the external knowledge comes in the form of general rules of inference that can be applied to increase target task performance. For example, given small circles, large circles, and small squares, can we learn to detect large squares? This can potentially be accomplished by learning a transformation from the source domain (small objects) to the target domain (large objects), and then applying that transformation to the labeled data in the source domain (small squares).

This work introduces the inductive transfer encoder (ITE) as a novel approach to the problem of cross-domain biometric comparisons. The ITE is an extension of the autoencoder that, given a sample in the source domain, attempts to reconstruct the sample in the target domain. Motivated by problem scenarios in keystroke biometrics, for which several public cross-domain datasets exist, the application of the ITE is shown to improve verification performance when the template samples are in the source domain and the genuine query samples are in the target domain. The

This research was supported in part by an appointment to the Postgraduate Research Participation Program at the U.S. Army Research Laboratory administered by the Oak Ridge Institute for Science and Education through an inter agency agreement between the U.S. Department of Energy and USARL.

rest of this paper is organized as follows. Section 2 reviews background material, Section 3 defines the cross-domain verification scenarios and motivates the development of an inductive transfer learning model, Section 4 defines the ITE, Section 5 contains experiment results, and Sections 6 and 7 contain a discussion and conclusions, respectively.

2. Background

2.1. Cross-domain challenges

Keystroke biometrics currently faces a several challenges, some of which can be attributed to cross-domain sample comparisons. We highlight some challenges here and refer the reader to [2] and [16] for complete surveys.

Typing behavior varies with keyboard device, as Crump et. al. have confirmed that differences in keystroke timing distributions and error rates can be observed across different device types [4]. In fixed-text and free-text input scenarios, Tappert et. al. have shown that identification accuracy depends on whether the template and query samples were collected on traditional desktop or laptop keyboards [15]. With a 36 subject population, identification accuracies upwards of 0.99 are achieved under ideal conditions, where the template and query samples are collected under the same input task and the same device type. When the template and query samples come from different domains, such as different device types or input tasks, the identification accuracies range from 0.50 to 0.90.

Impairment to normal typing behavior presents another challenge in which verification performance can be expected to degrade. This situation arises when either the subject's template or query samples are collected under conditions in which normal typing behavior is impaired. The *One Handed Keystroke Biometric Identification Competition* presented this challenge in the form of a biometric competition [13]. Participants were provided a labeled dataset with normal typing behavior and an unlabeled dataset that contained samples typed with one hand in addition to normal two-handed samples. Identification accuracies of the one-handed samples were about half of those obtained for the two-handed samples by all of the top-ranking participants. Additionally, the faster-typed samples were generally easier to classify than the slower-typed samples, with a -0.38 correlation between median press-press latency and identification accuracy.

Finally, changes in a subject's typing proficiency over time may ultimately determine keystroke biometric system performance. In a longitudinal study, Gentner confirms that the distribution of timings of a transcription typist evolves as typing proficiency increases [5]. This phenomenon is perhaps more apparent for short fixed-text sequences, such as passwords and PINs. As the subject repeats typing the sequence, the practice level increases and keystroke timings

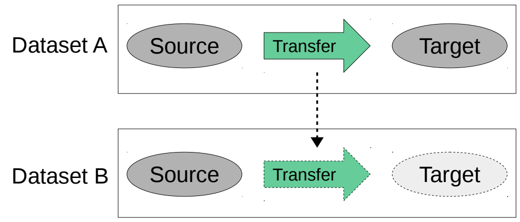


Figure 1: Inductive transfer learning model. A general rule is learned from pairwise correspondences between source and target domain samples in Dataset A. The model is applied to Dataset B, which only contains samples in the source domain. Samples in Dataset B are mapped to the target domain using the general rule learned in Dataset A.

may differ from when the sequence was completely novel to the subject. Keystroke template update mechanisms are usually employed to combat this issue [6]. Giot et. al. establish a relationship between the number of practice repetitions and the genuine match score [7]. The practice level of the impostor is also a determinant of system performance. Killourhy et. al. consider six factors that affect keystroke biometric verification performance, with impostor practice being one of them, and conclude that the effects of impostor practice are mitigated when a template update strategy is employed [11]. In Section 3, we will demonstrate how the practice levels of the template, genuine, and impostor subjects interplay to affect verification performance.

2.2. Transfer learning

Inductive transfer learning attempts to improve the performance of a target task by utilizing general rules of inference derived from a source task with labeled data [14]. This is in contrast to transductive transfer learning, which utilizes specific rules derived from both target and source data to solve a common task. A transductive transfer learning model has to be retrained for every target dataset whereas an inductive transfer model does not.

Given a labeled dataset with pairwise correspondences between samples in the source domain to samples in the target domain, an inductive learning model may attempt to learn a general transformation that can be applied to map samples from the source domain to the target domain. This scenario is demonstrated in Figure 1. The general rule can then be applied to another dataset for which only labeled samples in the source domain exist.

There are many real-world biometric scenarios in which it would be preferable to have samples in some target domain, yet only source domain samples are available. For example in keystroke biometrics, a system may attempt to verify a subject on a laptop keyboard although the template was collected on a desktop keyboard. Similarly, it may attempt to verify a subject who has practiced typing a pass-

	Domain		
Scenario	Template	Genuine	Impostor
A	Source	Source	Source
B	Source	Source	Target
C	Source	Target	Source
D	Source	Target	Target

Table 1: Cross-domain biometric verification scenarios in which template samples belong to the source domain and genuine and impostor samples belong to either the source or target domain. When transferring the template samples from the source domain to the target domain, we would expect a performance decrease in Scenarios A and B and a performance increase in Scenarios C and D, shown in bold.

	Practice level (Repetitions)			
	Template	Genuine	Impostor	EER
A	L (1)	L+H (2-200)	L (1-5)	0.412 (0.151)
B	L (1)	L+H (2-200)	H (396-400)	0.264 (0.134)
C	L (1)	H (201-400)	L (1-5)	0.511 (0.169)
D	L (1)	H (201-400)	H (396-400)	0.365 (0.183)
BC	L+H (1-200)	H (201-400)	L (1-5)	0.153 (0.093)
AD	L+H (1-200)	H (201-400)	H (396-400)	0.206 (0.123)

Table 2: Verification performance depends on subject practice level. Sample repetitions correspond to the number of times the subject typed the input sequence. Per-subject EER is obtained by a Manhattan distance anomaly detector. Scenarios A-D are one-shot learning scenarios in which the first repetition is used as the template. Scenarios BC and AD contain both low and high practice levels in the template, comprised of the first 200 repetitions. Scenarios BC corresponds to the validation procedure in [12].

word many times using a template that was collected when the password was novel to the subject. Finally, the system may attempt to verify a subject using a template that was collected under degraded conditions, such as only being able to type with one hand. There are countless other cross-domain scenarios that arise when biometric verification is applied in the real world. The verification performance of the system depends on which domain each sample (template, genuine, and impostor) belongs to.

3. Motivation

In a biometric verification scenario, given the template samples originate from a source domain, there are exactly 4 different scenarios in which the genuine and impostor samples can originate from either the source or target domain. These scenarios are shown in Table 1.

Scenario A in Table 1 is the ideal condition traditionally assumed by many verification systems, in which all the samples belong to the same domain. Scenario B is also ideal since samples in the target domain will generally be different than samples in the source domain, further decreasing the similarity between impostor samples and the template. Scenarios C and D are of interest since under these conditions a transfer learning model may improve performance. In Scenarios C and D, the template can be transferred to the target domain and made more similar to the genuine samples, potentially increasing verification performance. Scenario C offers the greatest room for improvement, since transferring the template from the source to the target domain will simultaneously increase similarity to the genuine samples and decrease similarity to the impostor samples. Scenario D is more difficult since transferring the template to the target domain can increase its similarity to both the genuine and impostor samples. Analogously, transferring template samples from the source to the target domain in Scenarios A and B can be expected to decrease verification performance since the templates are being made less similar to the genuine samples.

This work evaluates two sets of domains in keystroke biometrics: levels of typing practice and subject impairment. The rest of this section provides motivating examples for each of the scenarios in Table 1 in which no transfer learning model is applied.

3.1. Practice levels

The CMU benchmark keystroke dataset contains keystroke timing vectors from 51 subjects typing the password .tie5Roanl followed by the Enter key [12]. Each subject typed 400 repetitions split up into 8 sessions separated by at least one day. The keystroke timing vectors contain the key-hold durations, press-press latencies, and release-press latencies, for a total of 31 timing features. Key press and key release event timestamps were obtained with a high-precision clock, accurate to within 200 microseconds.

Since the CMU dataset contains many repetitions for each subject, the samples span various levels of typing practice. This ranges from a low practice level, in which the sequence is completely novel during the first attempt, to a high practice level after several hundred repetitions are completed. Given the template samples are collected at a low practice level, the genuine and impostor query samples can have either a low or high practice level. In a one-shot learning scenario in which the first sample is used as the template, verification accuracies are determined for each of the cross-domain scenarios in Table 1. Table 2 shows the per-subject equal error rate (EER) obtained by a Manhattan distance anomaly detector [12]. Using a cutoff of 50, i.e., repetitions 1-50 (the first session) are considered low practice, and sessions 51-400 are considered high practice, the domain of each group is labeled as either L=low practice, H=high practice, or L/H=mix of low and high practice samples.

	Typing hands			
Scenario	Template	Genuine	Impostor	EER
A	One	One	One	0.186 (0.119)
B	One	One	Both	0.073 (0.085)
C	One	Both	One	0.596 (0.178)
D	One	Both	Both	0.419 (0.191)

Table 3: Verification performance depends on subject impairment (typing with one or both hands). In all scenarios, EER is determined using Manhattan distance, 1 template sample, 9 or 10 genuine samples, and 109 (from every other subject) impostor samples.

The templates in Scenarios BC and AD are comprised of the first 200 repetitions form each subject, and contain samples in both the source and target domain. Note that Scenario BC is exactly the validation procedure described in [12], and the EER of Scenario BC corresponds to Detector 8 (Manhattan distance) in [12]. The results in Table 2 indicate that lower EER is achieved using template and genuine samples with high practice levels and impostors with low practice levels. In the real world, enrollment may be performed at a time when the subject has a low practice level, which corresponds to Scenarios A-D.

3.2. Typing hands

The GREYC-NISLAB keystroke dataset contains keystroke timing vectors from 110 subjects typing 5 different passphrases with between 17 and 24 characters [10]. Each subject typed each passphrase for 10 repetitions using only their dominant hand and 10 repetitions using both hands, for a total of 20 repetitions per subject. The keystroke timing feature vectors consist of the four types of latencies, including: press-press, release-release, press-release, and release-press. The dataset was collected on standard desktop keyboards, with a mix of AZERTY and QWERTY keyboard layouts.

This dataset presents a unique opportunity to recognize subjects with impaired typing during short fixed-text entry. Typing impairments have been the focus of some recent work in keystroke biometrics [13]. A robust keystroke biometric verification system must be able to deal with real-world scenarios, such as typing with only one hand. This task has proven to be difficult, as typing behavior changes when using only one hand.

Implementing the scenarios from Table 1, the template of each subject is comprised of a single one-handed sample, and the genuine and impostor query samples are either one- or two-handed. Table 3 shows the average EER per subject over all passphrases for each scenario. The results demonstrate that the verification task is most difficult when the genuine samples are in a different domain than the template.

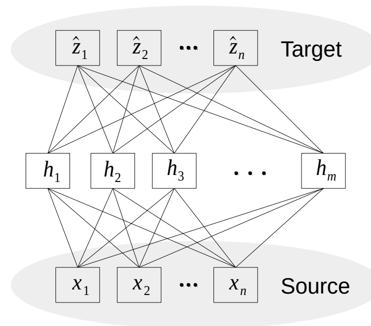


Figure 2: The transfer encoder learns a transformation between the source domain and target domain by trying to reconstruct source domain samples in the target domain.

Verification performance of Scenarios C and D could be improved by a function that transforms one-handed templates to be more similar to two-handed genuine samples. This serves as motivation for the development of the model introduced in the next section, which learns such a transformation from an independent dataset.

4. Inductive transfer encoder

The basic autoencoder attempts to reconstruct its input from an intermediary representation by applying an encode function followed by a decode function [3, 9]. The encode function, f_θ, transforms input $\mathbf{x}$ into hidden representation $\mathbf{h} = f_\theta(\mathbf{x})$, where θ is a parameter vector. A decode function $g_{\theta'}$ is then applied to the hidden representation to obtain the reconstructed input, $\widehat{\mathbf{x}} = g_{\theta'}(\mathbf{h})$. Realized as a neural network, the autoencoder is typically implemented with tied weights such that $\theta = \theta'$ in order to restrict the parameter space and act as a kind of regularization.

The inductive transfer encoder (ITE) is proposed in this work as a mechanism for transferring samples from the source domain to the target domain. The ITE is similar in respect to the basic autoencoder, except that given input $\mathbf{x}$ in the source domain, it attempts to reconstruct $\mathbf{x}$ in the target domain. This is accomplished through a set of pairwise correspondences between domains. The pairwise correspondences form a bipartite graph between samples in the source domain and samples in the target domain. Each pair consists of sample $\mathbf{x}$ in the source domain and the corresponding sample $\mathbf{z}$ in the target domain. The rest of this sections defines the encode and decode functions for the ITE and several strategies that can be used to establish the pairwise correspondences.

4.1. Encode and decode functions

The encode function maps the input to a single hidden layer,

$$\mathbf{h} = f_\theta(\mathbf{x}) = \tanh(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (1)$$

where $\mathbf{x}$ is the length-n input in the source domain, $\mathbf{W}$ is a $(n \times m)$ weight parameter matrix, $\mathbf{b}$ is a length-m bias parameter vector, and $\tanh$ is the hyperbolic tangent function. The decode function is given by

$$\widehat{\mathbf{z}} = g_{\theta'}(\mathbf{y}) = \tanh\left(\mathbf{W}^\top \mathbf{h} + \mathbf{c}\right) \quad (2)$$

where $\widehat{\mathbf{z}}$ is the length-n reconstructed output in the target domain, $\mathbf{W}^\top$ is the $(m \times n)$ tied-weight parameter matrix, and $\mathbf{c}$ is a length-n output bias parameter vector. The complete set of model parameters consists of $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$. The loss function is the squared error between the actual and reconstructed output,

$$L(\mathbf{z}, \widehat{\mathbf{z}}) = \|\mathbf{z} - \widehat{\mathbf{z}}\|^2 \quad (3)$$

where $\mathbf{z}$ is the target domain sample corresponding to $\mathbf{x}$.

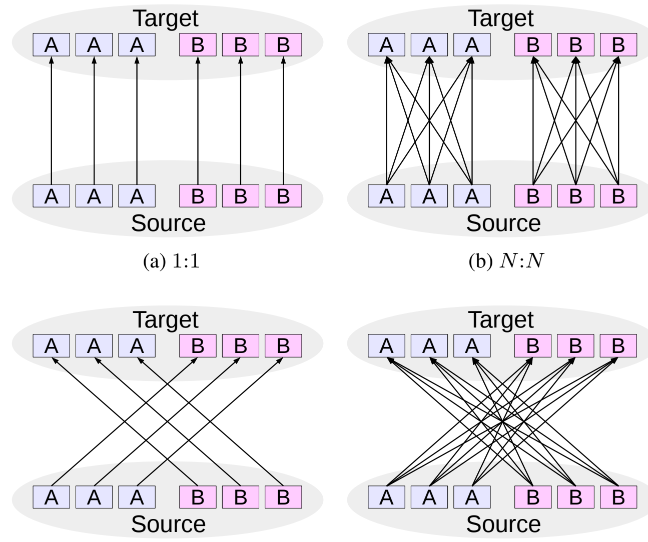
The ITE is agnostic to the particular encode, decode, and loss functions used. Instead, the ITE is characterized by reconstructing source domain samples in the target domain. Other implementations are possible, such as using multiple hidden layers, different (nonlinear) activation functions, or different loss functions. The described ITE topology is shown in Figure 2, where the size of the hidden layer, m, is a hyperparameter. Since it is generally true that $\mathbf{x} \neq \mathbf{z}$, an overcomplete representation can be obtained in the hidden layer. This allows choices of $m \geq n$ without the need for additional regularization terms to avoid learning the identity function.

4.2. Bipartite strategies

The question of how to establish the pairwise correspondences still remains. Given a dataset with samples in both the source and target domain, there are many ways the pairwise correspondences can be established. In this work, we evaluate four different bipartite strategies. Each strategy defines a bipartite graph between samples in the source domain and the target domain. Each edge in the graph consists of sample pair $\{\mathbf{x}, \mathbf{z}\}$, where $\mathbf{x}$ is in the source domain and $\mathbf{z}$ is in the target domain. The set of all pairs is used to determine the parameters of the ITE.

The 1-to-1 (1:1) strategy links each sample in the source domain to a single sample from the same subject in the target domain. The class label (subject) remains invariant when moving across domains, i.e, every $\{\mathbf{x}, \mathbf{z}\}$ pair of samples both belong to the same subject. With N samples from each of M subjects in each domain, this strategy produces MN correspondence pairs.

The N-to-N (N:N) strategy links each sample in the source domain to every other sample from the same subject in the target domain. Similar to the 1:1 strategy, the subject remains invariant across domains. With N samples from each of M subjects in each domain, this strategy produces MN^2 correspondence pairs.



(a) 1:1 (b) N:N

(c) 1×1 (d) $N \times N$

Figure 3: Bipartite strategies used to establish pairwise correspondences between the source domain and target domain. Each edge corresponds to one pair presented to the ITE for parameter estimation. The 1:1 and N:N strategies preserve subject class labels while the links in the 1×1 and $N \times N$ strategies do not.

The 1-cross-1 (1×1) strategy links each sample in the source domain to a single sample from a different subject in the target domain. Different from the preceding two strategies, the subject does not remain invariant when moving from the source to the target domain. Each pair $\{\mathbf{x}, \mathbf{z}\}$ consist of sample $\mathbf{x}$ in the source domain from Subject A and sample $\mathbf{z}$ in the target domain from Subject B. With N samples from each of M subjects in each domain, this strategy produces MN correspondence pairs.

The N-cross-N ($N \times N$) strategy links each sample in the source domain to every other sample from a different subject in the target domain. Similar to the 1×1 strategy, the subject does not remain invariant when moving across domains. With N samples from each of M subjects in each domain, this strategy produces MN^2 correspondence pairs.

The four bipartite strategies are summarized in Figure 3, showing 2 subjects with 3 samples in each domain. Each edge between the source and target domain corresponds to one pair presented to the ITE for parameter estimation. The edges in the 1:1 and N:N strategies preserve class labels while the edges in the 1×1 and $N \times N$ strategies do not. The 1×1 and $N \times N$ strategies are evaluated to determine whether subject-invariant transformations are captured by the 1:1 and N:N strategies.

5. Experiments

Two sets of experiments are performed to evaluate verification performance for each of the cross-domain transfer learning scenarios (Table 1) and bipartite strategies (Figure 3) on two cross-domain problems. The ITE is implemented in Python using tensorflow [1], a modular machine learning framework. Parameters are determined by stochastic gradient descent with learning rate 0.01, terminated after 1000 epochs. Weight parameters use a Xavier initialization [8], drawing from a uniform distribution scaled by the number of inputs, specifically $\mathcal{U}\left(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$, and biases are initialized to 0. Every $\{\mathbf{x}, \mathbf{z}\}$ pair is examined during one epoch, i.e, mini-batches are not used. The low practice $\rightarrow$ high practice experiments used an ITE with $m = 300$ (i.e., a hidden layer with 300 units), and the one hand $\rightarrow$ both hands experiments used an ITE with $m = 1000$. These values were chosen from a small range of values between 100 and 10000 such that verification performance remained relatively stable for larger m. Source code to reproduce experiment results is located at https://github.com/vmonaco/transfer-encoder.

Verification performance is reported as the equal error rate (EER), the point on the receiver operating characteristic (ROC) curve where the false acceptance rate (FAR) and false rejection rate (FRR) are equal. Confidence intervals are obtained through a Monte Carlo cross validation procedure. In each fold, the samples from M subjects are held out to determine the parameters of the ITE under a particular bipartite strategy, and the samples from the remaining subjects are used to validate the model. The samples from the remaining subjects are split up into template, genuine, and impostor query samples as described below. The EER is determined with a Manhattan distance anomaly detector [12] before and after applying the ITE to the template samples for each bipartite strategy. This process is repeated 10 times. Note that ITE parameter estimation is performed using a subset of data that is independent from that used to determine the EER. This procedure evaluates whether the ITE can capture a general transformation between the source and target domain.

5.1. Low practice $\rightarrow$ high practice

The CMU benchmark keystroke dataset [12] is used to evaluate the ITE in a task where template samples are transferred from the source domain (low practice) to the target domain (high practice). The source domain consists of samples from the first session (repetitions 1-50) and the target domain is the last session (repetitions 351-400). In each fold, samples from 25 subjects are held out to train a 300-hidden unit ITE, and samples from the remaining 26 subjects are used to determine the EER before and after applying the ITE to the templates.

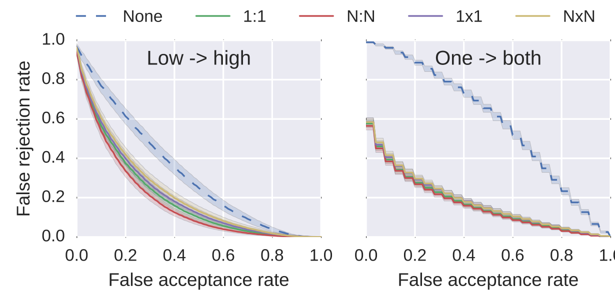


Figure 4: Cross-domain Scenario C ROC curves (95% CI).

Experiment results are shown in Table 4, and ROC curves for cross-domain Scenario C are shown in Figure 4. In Scenarios A-D the first repetition of each subject is used as the template, i.e., they are one-shot learning tasks. Low+high practice genuine query samples consist of repetitions 2-200 from the same subject and high practice genuine query samples include repetitions 201-400. Low practice impostor samples are comprised of the first ten repetitions of every other subject and high practice impostor samples are comprised of the last ten repetitions. Thus, in determining the EER for each subject there is 1 template, approximately 200 genuine query samples, and 250 impostor query samples. The average EER is obtained over 26 subjects and 10 folds. Scenario BC is similar to the validation procedure in [12], except with only 26 subjects. In Scenarios BC and AD, the templates consist of both low and high practice samples.

5.2. One hand $\rightarrow$ both hands

The GREYC-NISLAB keystroke dataset [10] is used to evaluate the ITE transferring template samples typed with only one hand to both hands. The source domain consists of 10 samples per subject typed with only the subject's dominant and the target domain contains 10 samples per subject typed normally with both hands. For each fold in the validation procedure described above, samples from 84 subjects are held out to train a 1000-hidden unit ITE, and samples from the remaining 26 subjects are used to determine the EER with and without transfer.

Table 5 shows the experiment results for verification Scenarios A-D, and ROC curves for cross-domain Scenario C are shown in Figure 4. In each scenario, a single one-handed sample is used as the template. The remaining 9 one-handed samples from the same subject comprise the genuine query samples in the source domain and the 10 two-handed samples from the same subject comprise the genuine query samples in the target domain. The source domain impostor query samples consist of a single one-handed sample from every other subject, and the target domain impostor query samples consist of a two-handed sample from every other

	Practice level			Bipartite strategy EER				
	Template	Genuine	Impostor	None	1:1	N:N	1×1	N×N
A	Low	Low+High	Low	0.272 (0.136)	0.228 (0.120)	**0.209 (0.106)**	0.237 (0.124)	0.253 (0.134)
B	Low	Low+High	High	**0.186 (0.117)**	0.222 (0.123)	0.206 (0.110)	0.237 (0.119)	0.253 (0.124)
C	Low	High	Low	0.363 (0.157)	0.254 (0.128)	**0.239 (0.112)**	0.268 (0.132)	0.280 (0.136)
D	Low	High	High	0.262 (0.154)	0.246 (0.128)	**0.233 (0.120)**	0.267 (0.125)	0.277 (0.122)
BC	Low+High	High	Low	0.119 (0.086)	0.064 (0.047)	**0.063 (0.047)**	0.074 (0.050)	0.080 (0.052)
AD	Low+High	High	High	0.139 (0.102)	0.119 (0.079)	**0.119 (0.077)**	0.137 (0.078)	0.141 (0.077)

Table 4: Verification performance in a transfer learning task with low and high practice levels. The EER is obtained before and after applying the ITE with four different bipartite strategies (see Section 4.2). Scenarios A-D are one-shot learning tasks, using only the first repetition from each subject as the template. The templates in Scenarios BC and AD contain a mix of low and high practice levels. Scenario BC is comparable to the validation procedure in [12].

	Typing hands			Bipartite strategy EER				
	Template	Genuine	Impostor	None	1:1	N:N	1×1	N×N
A	One	One	One	**0.172 (0.121)**	0.225 (0.131)	0.220 (0.129)	0.228 (0.133)	0.235 (0.134)
B	One	One	Both	**0.069 (0.096)**	0.430 (0.219)	0.421 (0.216)	0.438 (0.201)	0.443 (0.209)
C	One	Both	One	0.573 (0.195)	0.217 (0.178)	**0.215 (0.176)**	0.229 (0.187)	0.231 (0.188)
D	One	Both	Both	0.390 (0.172)	0.363 (0.173)	**0.356 (0.171)**	0.379 (0.173)	0.379 (0.175)

Table 5: Verification performance in a transfer learning task with one and two typing hands. The EER is obtained before and after applying the ITE with four different bipartite strategies (see Section 4.2). All scenarios are one-shot learning tasks where the template consists of a single one-handed sample. Genuine and imposter query samples can be either one- or two-handed.

subject. To summarize, the EER for each is determined by 1 template, 9 genuine query samples, and 25 impostor query samples. This process is repeated for each passphrase in addition to the 10 folds. The average EER is obtained over 26 subjects, 10 folds, and 5 passphrases.

6. Discussion

The results in Tables 4 and 5 demonstrate the difficulty in comparing samples across domains. An increase in verification accuracy is observed in Scenarios C and D after the templates are transferred to the target domain using an ITE trained on an independent dataset. This supports the hypothesis that there exist general mappings between domains. The mapping can be learned by the ITE and subsequently applied to other samples in the source domain.

The ability to transfer low practice samples to high practice samples suggests that there is a general rule by which keystroke templates age as the sequence becomes practiced by the subject. The ITE presents a safe alternative to keystroke template update mechanisms since query samples are never incorporated into the subject's template, and thus, is not vulnerable to template-poisoning attacks from impostors [18].

In Scenarios A and B, the EER usually increases after each transfer strategy is applied. This is expected, since a template transfer in Scenario A will decrease similarity with the genuine samples and a template transfer in Scenario B will simultaneously decrease similarity with the genuine

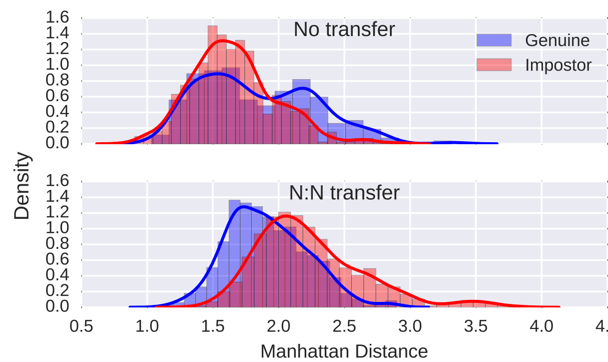


Figure 5: Scenario D genuine and impostor score distributions from a single subject before and after the template is transferred to the target domain.

and increase similarity with the impostor samples. The opposite effect is achieved in Scenarios C and D. A look at the distance distributions of a single subject from the low practice → high practice experiments confirms this, shown in Figure 5. The distribution of impostor distances is shifted away from the distribution of genuine distances after the ITE is applied, trained with the N:N bipartite strategy. It is also the case that the EERs of the 1:1 and N:N strategies are lower than the 1×1 and N×N strategies. This suggests that the 1:1 and N:N bipartite strategies preserve subject-dependent characteristics, although the improvements are mostly marginal.

7. Conclusions

This work defined several cross-domain biometric verification scenarios and identified some cross-domain challenges in keystroke biometrics. The ITE was proposed as a mechanism for transferring samples from a source domain to a target domain. The ITE is an inductive transfer learning model that attempts to learn a general transformation across domains from a set of pairwise correspondences between samples in the source and target domains. Samples in the source domain are encoded into a hidden feature representation and then decoded in the target domain. The model was then applied to a task where only labeled samples in the source domain are available.

The ITE can alternatively be interpreted as a denoising autoencoder in which noise is introduced by the source domain. The denoising autoencoder learns a robust representation of its input by attempting to reconstruct a sample after it has undergone some corruption process [17]. Likewise, the ITE learns to reconstruct a sample in the target domain after it has undergone the effects of the source domain.

There are several directions for future research. Prominently, the issue of *when* to transfer still remains. In this work, we have not addressed the issue of when to transfer a sample from the source domain and assumed that the domain of each sample (template, query, impostor) is known. In practice, this will not be the case, and a decision of whether to transfer a sample needs to be made. This is also an open question in transfer learning [14].

A deeper analysis of bipartite strategies should be performed. With N samples from each of M subjects in each domain, there are 2^{MN} ways to construct the bipartite graph. This work examined only a small subset of possibilities, given by the 1:1, N:N, 1×1, and $N\times N$ strategies. Alternative strategies might use some heuristic, such as similarity between samples across domains, to construct the bipartite graph. There are also numerous cross-domain scenarios in which the ITE can potentially be used to increase verification accuracy, such as comparing samples collected from different sensors.

References

[1] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] S. P. Banerjee and D. L. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[3] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[4] M. J. Crump and G. D. Logan. Warning: This keyboard will deconstruct - the role of the keyboard in skilled typewriting. *Psychonomic bulletin & review*, 17(3):394–399, 2010.

[5] D. R. Gentner. Keystroke timing in transcription typing. In *Cognitive aspects of skilled typewriting*, pages 95–120. Springer, 1983.

[6] R. Giot, B. Dorizzi, and C. Rosenberger. Analysis of template update strategies for keystroke dynamics. In *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2011 IEEE Workshop on*, pages 21–28. IEEE, 2011.

[7] R. Giot, C. Rosenberger, and B. Dorizz. Can chronological information be used as a soft biometric in keystroke dynamics? In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 7–10. IEEE, 2012.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010.

[9] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pages 3–3, 1994.

[10] S. Z. S. Idrus, E. Cherrier, C. Rosenberger, and P. Bours. Soft biometrics database: A benchmark for keystroke dynamics biometric systems. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*, pages 1–8. IEEE, 2013.

[11] K. Killourhy and R. Maxion. Why did my detector do that?! In *Recent Advances in Intrusion Detection*, pages 256–276. Springer, 2010.

[12] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE, 2009.

[13] J. V. Monaco, G. Perez, C. C. Tappert, P. Bours, S. Mondal, S. Rajkumar, A. Morales, J. Fierrez, and J. Ortega-Garcia. One-handed keystroke biometric identification competition. In *Biometrics (ICB), 2015 International Conference on*, pages 58–64. IEEE, 2015.

[14] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[15] C. C. Tappert, M. Villani, and S.-H. Cha. Keystroke biometric identification and authentication on long-text input. *Behavioral biometrics for human identification: Intelligent applications*, pages 342–367, 2009.

[16] P. S. Teh, A. B. J. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013, 2013.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[18] Z. Wang, A. Serwadda, K. S. Balagani, and V. V. Phoha. Transforming animals in a cyber-behavioral biometric menagerie with frog-boiling attacks. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 289–296. IEEE, 2012.