Doctoral Dissertation

# Research on collaborative machine learning with a human expert for supporting network operations

Hidetoshi Kawaguchi

Supervisor: Shogo Okada

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

March, 2023

**Abstract**

Communication networks have become indispensable to people's lives. In this age of smartphones, the Internet can be regarded as an infrastructure for daily life and electricity, water, and gas. Companies that provide such communication services to users struggle daily to ensure the stable communication networks.

Network operations experts are needed to ensure the stable communication networks. However, with new technologies such as the Internet of things (IoT) and 5G, the burden on the experts who manage them continues to increase. The work of network operations experts is diverse and includes many decision-making tasks.

We focus on the intrusion detection and prevention system (IDPS) signature classification task in network operations. IDPSs monitor network systems and take actions such as logging, notification, and blocking when malicious communications are detected. This dissertation focuses on a type of IDPS that performs detection based on pattern files of malicious communications such as signatures. The signatures are distributed periodically by the IDPS developers, similar to a subscription service. Network operations experts determine the importance level ("low", "medium", or "high") of each signature to set IDPS actions. For example, if the importance of a signature is "high", the action is "blocking"; if it is low, the action is "logging".

Determining the importance level of a signature for setting it up in the IDPS is a burden for the expert. While expert-designed if-then rule scripts can automatically determine some signatures, the remaining signatures must be determined manually by experts based on elements in the signatures, articles on the Internet, and their own experience. This manual decision-making process takes some time. In addition to time consuming issues, it takes sufficient knowledge and experience to determine the importance level, and there are not many experts who have such knowledge and experience. In other words, the cost of hiring experts is high. Therefore, determining the importance of a signature, which requires an expert's time, is also a significant cost to the network company.

IDPS signature classification is an important task, but there has yet to be research to automate it. We must recognize the seriousness of classification errors that can result from automation, as is the case in the medical field. Signature misclassification leads to IDPS misconfiguration. Misconfiguration of IDPS can cause security incidents, such as missing malicious communications and false interceptions of regular communications. Security incidents should be avoided because they damage public trust in the organization operating the network system. Hence, it is not practical to automate all signature classifications. In order to automate

classification while reducing classification errors, a framework for efficient classification in cooperation with humans, such as checking with experts as necessary, is required.

This dissertation aims to formulate IDPS signature classification as a machine learning problem for the first time and to build and evaluate a system that cooperates with experts to classify signatures. To achieve this goal, we addressed three problems.

First, there are no publicly available datasets for machine learning signature classification. In other words, they need to prepare for the prerequisites of the research. Several reasons make signature datasets difficult to make publicly available: Many of the signatures are distributed by IDPS developers, but they cannot be redistributed under license; Publishing the signatures and their labels may lead to the outside world guessing about the sensitive information of the IDPS configuration. We collected the three datasets used in this research in cooperation with several experts in actual network operations organizations. These are real datasets consisting of signatures that experts actually set in the IDPS. Experts classify some signatures by predefined if-then rules. An if-then rule returns a label of "low", "medium", "high", or "unknown" importance based on keyword matching of the elements in the signature. Two datasets, the automatically annotated dataset (AAD) and the manually annotated dataset (MAD), were collected. AAD consists of $4,465$ signatures automatically labeled by expert-designed if-then rule scripts. MAD consists of $1,300$ signatures that could not be classified by the if-then rule scripts and were manually labeled by the experts. Next, we collected a time-series manually annotated dataset (TMAD) consisting of $7,577$ signatures that were manually labeled and time-stamped with the date and time of distribution. Both labels of signatures were determined after consultation with several experts. This research is based on these three datasets.

Second, to classify IDPS signatures by machine learning, it is necessary to search for an effective feature extraction method. We propose three features based on the expert's knowledge, with interpretability to clarify the expert's criteria. We first design two types of features, symbolic features (SFs) and keyword features (KFs), which are used in keyword matching for the if-then rules. Next, we design web information and message features (WMFs) to capture the properties of signatures that do not match the if-then rules. The WMFs are extracted as term frequency-inverse document frequency (TF-IDF) features of the message text in the signatures. The features are obtained by web scraping from the referenced external attack identification systems described in the signature. The effectiveness of the proposed features is evaluated in experiments with AAD and MAD. In the experiment, the classification models with proposed features are evaluated from two perspectives: classification accuracy and reject option (RO) performance. In

both cases, the combined SFs and WMFs performed better than the combined SFs and KFs. We also show that using an ensemble of neural networks (deep ensembles; DE) improves the performance of the RO. An analysis shows that experts refer to natural-language elements in the signatures and information from external information systems on the Internet.

Third, if a fully automated machine learning model replaces the IDPS signature classification task, there is a risk of missing critical classification errors. It is also necessary to entrust experts with decisions that have a high risk of error by signature classification models. Therefore, it is important to establish a method for humans and the system to cooperate in setting up and classifying data. In addition, to actually use machine learning, it is necessary to cope with high annotation costs and domain shifts caused by signatures created to keep up with new cyber attacks. In this dissertation, we propose a system based on active learning in cooperation with experts to overcome three challenges: (a) security incidents caused by classification errors, (b) high annotation costs, and (c) classification accuracy decrease due to domain shifts. The proposed system includes an IDPS signature classification model and periodically classifies the received signatures in cooperation with an expert. The uncertainty sampling is used as an acquisition function to preferentially transfer signatures with a high risk of misclassification to the expert. The signatures are sorted by uncertainty sampling; some are transferred to experts, and the rest are automatically classified. The experts classify the transferred signatures and add them to the training dataset, and the classification model is retrained. After training, the new signatures that have not yet been labeled are classified. The proposed system executes this workflow each time it receives signatures. Uncertainty estimation methods in deep learning, such as Monte Carlo dropout (MC-Dropout) and DE, are also incorporated to identify signatures at high risk of misclassification accurately. Experiments are conducted on the TMAD to evaluate the proposed system in a simulation case. An analysis is then performed by comparing several variants of the proposed system. The results show that the system with MC-Dropout performs best. We also show that this variation has two effects: it transfers more samples with "medium" importance to the experts and mitigates imbalances in the training dataset.

As described above, in this dissertation, we collected IDPS signature datasets that are difficult to make public, and proposed features for machine learning classification of IDPS signatures and an active learning-based system to cooperate with experts. The proposed system enables accurate identification of IDPS signatures and contributes to reducing fatal classification errors, which are problematic in practical applications. Analysis using the proposed features identifies the elements in the signatures that are important to experts when classifying signatures. Identifying the important factors to experts can provide helpful information for other

machine learning and non-machine learning approaches to signature classification. The proposed system procedures are widely applicable not only to signatures. There are other tasks in network operations where data are generated periodically and classified by experts. For example, software vulnerability information, such as common vulnerabilities and exposures (CVE), is released periodically, and experts may decide whether to classify this information as necessary. In this dissertation, task sharing is considered collaboration, but interaction with machine learning systems and education of novices using them are also examples of collaboration. The realization of such collaborations is future works for machine learning technology to support network operations. We hope that the ideas and evaluation results in this dissertation will help solve signature classification problems as well as other tasks.

**Keywords:** machine learning, IDPS, signature, active learning, uncertainty estimation, reject option.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **AAD** | Automatically Annotated Dataset. |
| **Adam** | Adaptive Moment Estimation. |
| **ARC** | Accuracy-Rejection Curve. |
| **AU-ARC** | Area Under the Accuracy-Rejection Curve. |
| **BACC** | Balanced Accuracy. |
| **BALD** | Bayesian Active Learning by Disagreement. |
| **BERT** | Bidirectional Encoder Representations from Transformers. |
| **CNN** | Convolutional Neural Network. |
| **CO-BACC** | Co-Balanced Accuracy. |
| **CVE** | Common Vulnerabilities and Exposures. |
| **DE** | Deep Ensembles. |
| **DNN** | Deep Neural Network. |
| **DT** | Decision Tree. |
| **IDPS** | Intrusion Detection and Prevention System. |
| **ITRF** | If-then Rule Feature. |
| **KF** | Keyword Feature. |
| **MAD** | Manually Annotated Dataset. |
| **MCF** | Manual Classification Feature. |
| **MC-Dropout** | Monte Carlo dropout. |
| **MLP** | Multilayer Perceptron. |
| **NB** | Naive Bayes. |
| **NLP** | Natural Language Processing. |
| **NVD** | National Vulnerability Database. |
| **OvR** | One-vs-Rest. |
| **ReLU** | Rectified Linear Unit. |
| **RF** | Random Forest. |
| **RO** | Reject Option. |
| **SF** | Symbolic feature. |
| **SVM** | Support Vector Machine. |
| **TF-IDF** | Term Frequency-Inverse Document Frequency. |
| **TMAD** | Time-series Manually Annotated Dataset. |
| **WMF** | Web Information and Message Feature. |

# Chapter 1

# Introduction

Communication networks have become indispensable to people's lives. In this age of smartphones, the Internet can be regarded as an infrastructure for daily life and electricity, water, and gas. Companies that provide such communication services to users struggle daily to ensure the stable communication networks.

One of the main assignments of a communication network company is to perform network operations. Network operations include many tasks, such as network device configuration, troubleshooting, new equipment installation, and network security operations. Communication network companies continue to pay operating costs to maintain a communication environment that has grown to a large scale and are trying to reduce costs while maintaining high communication quality.

Communication network facilities are supported by operators daily. In this research, network operators are referred to as experts because they require advanced knowledge and practical experience. Specialized knowledge and experience are necessary to operate a growing computer network properly. Experts perform a wide variety of tasks, but many of them involve making decisions on given data.

Reducing the burden of experts contributes to the cost reduction of whole network operations. The costs of hiring experts are high because of their expertise, and it is not easy to hire them. Reducing human resource costs is a pressing issue for communications network companies.

We focus on the intrusion detection and prevention system (IDPS) signature classification task in network operations. The figure 1.1 shows an overview of IDPSs. IDPSs monitor network systems and take actions such as logging, notification, and blocking when malicious communications are detected. This dissertation focuses on a type of IDPS that performs detection based on pattern files of malicious communications such as signatures [1, 2, 3]. The signatures are distributed periodically by the IDPS developers, similar to a subscription service.

Since each IDPS user has a different security policy, the IDPS action when a signature is matched must be configured for each distributed signature. Depending

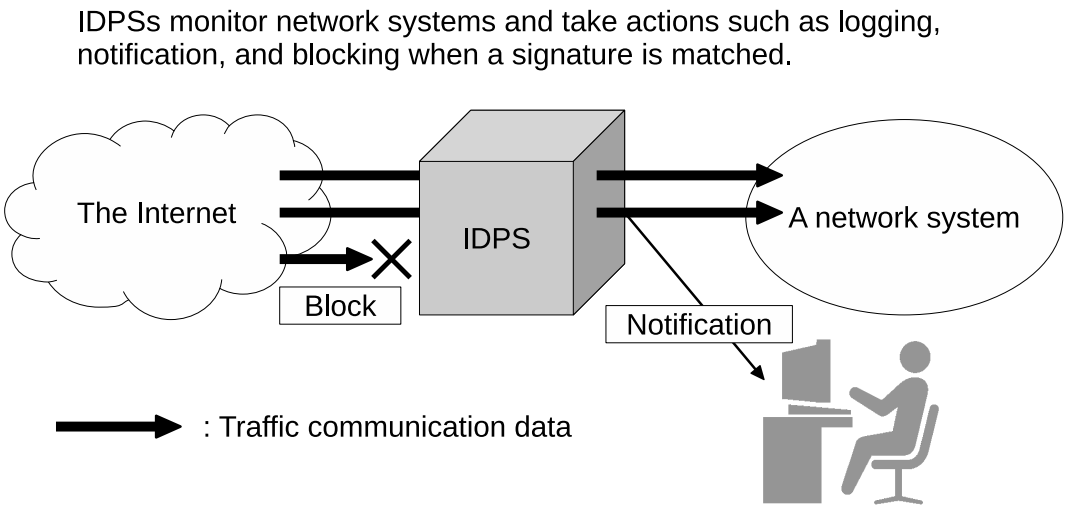


Figure 1.1: Intrusion detection and prevention system (IDPS)

on the hardware and software that make up the network system being monitored by IDPS, there are differences in the malicious communications that damage the system. For example, malicious communications that should be blocked in one organization may not need to be blocked in another. IDPS developers set a default action for each signature. However, for the above reasons, it is necessary to set an action for each signature that fits the organization.

Network operations experts determine the importance level ("low", "medium", or "high") of each signature to set IDPS actions. Figure 1.2 shows an overview of the importance determination made by experts on signatures and the IDPS settings involved. For example, if the importance level of a signature is "high", the action is "blocking"; if it is "low", the action is "logging". This importance level determination process requires expertise and should be considered a significant cost of network operations. In general, the experts classify signatures somewhat automatically. First, the experts classify signatures using an if-then rule script coded by them. The if-then rule returns an importance label or an "unknown" label according to the results of keyword matching on the elements in a signature. The experts then manually classify signatures that are determined to be "unknown" by the if-then rule.

Determining the importance level of a signature for setting it up in the IDPS is a burden for the expert. While expert-designed if-then rule scripts can automatically determine some signatures, the remaining signatures must be determined manually by experts based on elements in the signatures, articles on the Internet, and their own experience. This manual decision-making process takes some time. In addition to time consuming issues, it takes sufficient knowledge and experi-

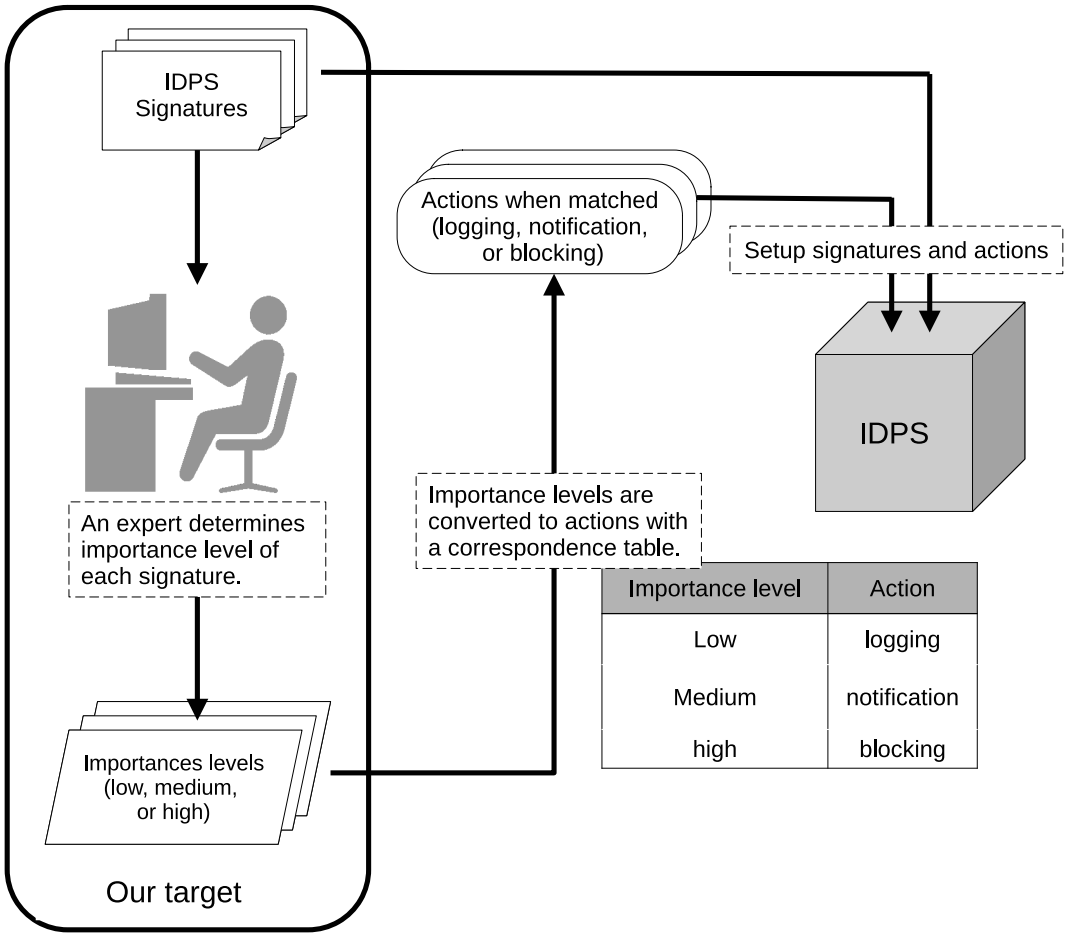


Figure 1.2: The importance determination made by experts on signatures and the IDPS settings involved

ence to determine the importance level, and there are not many experts who have such knowledge and experience. In other words, the cost of hiring experts is high. Therefore, determining the importance of a signature, which requires an expert's time, is also a significant cost to the network company.

IDPS signature classification is an important task, but there has yet to be research to automate it. We must recognize the seriousness of classification errors that can result from automation, as is the case in the medical field. Signature misclassification leads to IDPS misconfiguration. Misconfiguration of IDPS can cause security incidents, such as missing malicious communications and false interceptions of regular communications. Security incidents should be avoided because they damage public trust in the organization operating the network system. Hence, it is not practical to automate all signature classifications. A collaborative classifi-

cation framework is required that allows experts to classify signatures as needed. Our goal is to develop technologies that enable systems to be helpful to humans while contributing to improving system performance without humans being aware of it.

If the determination process of experts is modeled, computers can perform the network operation task, thereby reducing their burden. Machine learning is a practical modeling approach. In recent years, machine learning has been actively applied and put to practical use in media fields, such as image processing [4, 5], natural language processing (NLP) [6], and speech processing [7]. Machine learning is also being applied to communication networks, such as traffic prediction, resource management, and security [8, 9], but compared to the media field, there are many unexplored aspects.

This dissertation aims to formulate IDPS signature classification as a machine learning problem for the first time and to build and evaluate a system that cooperates with experts to classify signatures. To achieve this goal, we addressed three problems. The three problems and their solutions are described below:

**1. Collecting labeled IDPS signature datasets**

*Problem:* There are no publicly available datasets for machine learning signature classification. In other words, they need to prepare for the prerequisites of the research. Several reasons make signature datasets difficult to make publicly available. Many of the signatures are distributed by IDPS developers, but they cannot be redistributed under license. Publishing the signatures and their labels may lead to the outside world guessing about the sensitive information of the IDPS configuration.

*Solution:* We work with experts from real network operating organizations to collect three datasets. The signatures in these datasets are the actual inputs to the IDPS, and the labels are determined in consultation with several experts for this research. We also describe the notations of signatures and the expert's signature classification procedure, which are the premise of this research. (Chapter 3)

**2. Feature engineering based on experts' knowledge**

*Problem:* To classify IDPS signatures by machine learning, it is necessary to search for an effective feature extraction method. Since the signatures in this research are represented in text format, we can input them into a large-scale language model of deep neural networks (DNNs), which have rapidly developed in recent years [10, 11, 12, 13, 14]. However, DNNs have a problem with interpretability [15]. We need to know the criteria for experts to determine the importance level of signatures. As a starting point for this research, it is desirable to be as interpretable as possible and to be able to

identify the elements of interest to the expert.

*Solution:* We propose three features based on the expert's knowledge, with interpretability to clarify the expert's criteria. We first design two types of features, *symbolic features (SFs)* and *keyword features (KFs)*, which are used in keyword matching for the if-then rules. Next, we design *web information and message features (WMFs)* to capture the properties of signatures that do not match the if-then rules. The WMFs are extracted as term frequency-inverse document frequency (TF-IDF) features of the message text in the signatures. The features are obtained by web scraping from the referenced external attack identification systems described in the signature. (Chapter 4)

**3. A machine learning system that cooperates with an expert**

*Problem:* If a fully automated machine learning model replaces the IDPS signature classification task, there is a risk of missing critical classification errors. It is also necessary to entrust experts with decisions that have a high risk of error by signature classification models. Therefore, it is important to establish a method for humans and the system to cooperate in setting up and classifying data. In addition, to actually use machine learning, it is necessary to cope with high annotation costs and domain shifts caused by signatures created to keep up with new cyber attacks. In summary, to automate real-world IDPS signature classification with machine learning models, we need to overcome the following three challenges

(a) *Security incidents caused by classification errors* - Incorrect IDPS configurations due to classification errors may cause a security incident that could be fatal to the organization.

(b) *High annotation costs* - Only a limited number of people can annotate signatures due to the need for expertise.

(c) *Classification accuracy decreases due to domain shifts* - New signatures may result in decreased accuracy because the distribution of the new signatures is different from that of the trained signatures.

*Solution:* In this dissertation, we propose a system based on active learning [16] in cooperation with experts. The uncertainty sampling is used as an acquisition function to preferentially transfer signatures with a high risk of misclassification to the expert. Uncertainty estimation methods from deep learning [17], such as Monte Carlo dropout (MC-Dropout) [18] and deep ensembles (DE) [19], are also incorporated to estimate signature uncertainty accurately. The proposed system overcomes the following three challenges (a)-(c). (Chapter 5)

A roadmap of this research is shown in Figure 1.3. Work 1 is positioned as a preparation for starting a machine learning study to classify IDPS signatures. Work 2 is the first machine learning study of IDPS signature classification using the datasets from Work 1. Work 3 is an extension study of Work 2 using the other dataset from Work 1.

This dissertation is organized as follows. Chapter 2 describes the related works of this research. Chapter 3 describes the tasks and signatures of the studied experts and the collected dataset as the common problem set for this dissertation. In Chapter 4, we propose a feature design for signature classification models based on experts' knowledge and report the evaluation results using real datasets. In Chapter 5, we propose a machine learning system that cooperates with an expert, and report the evaluation results using a time-stamped dataset collected over two years. Chapter 6 presents our concluding remarks.

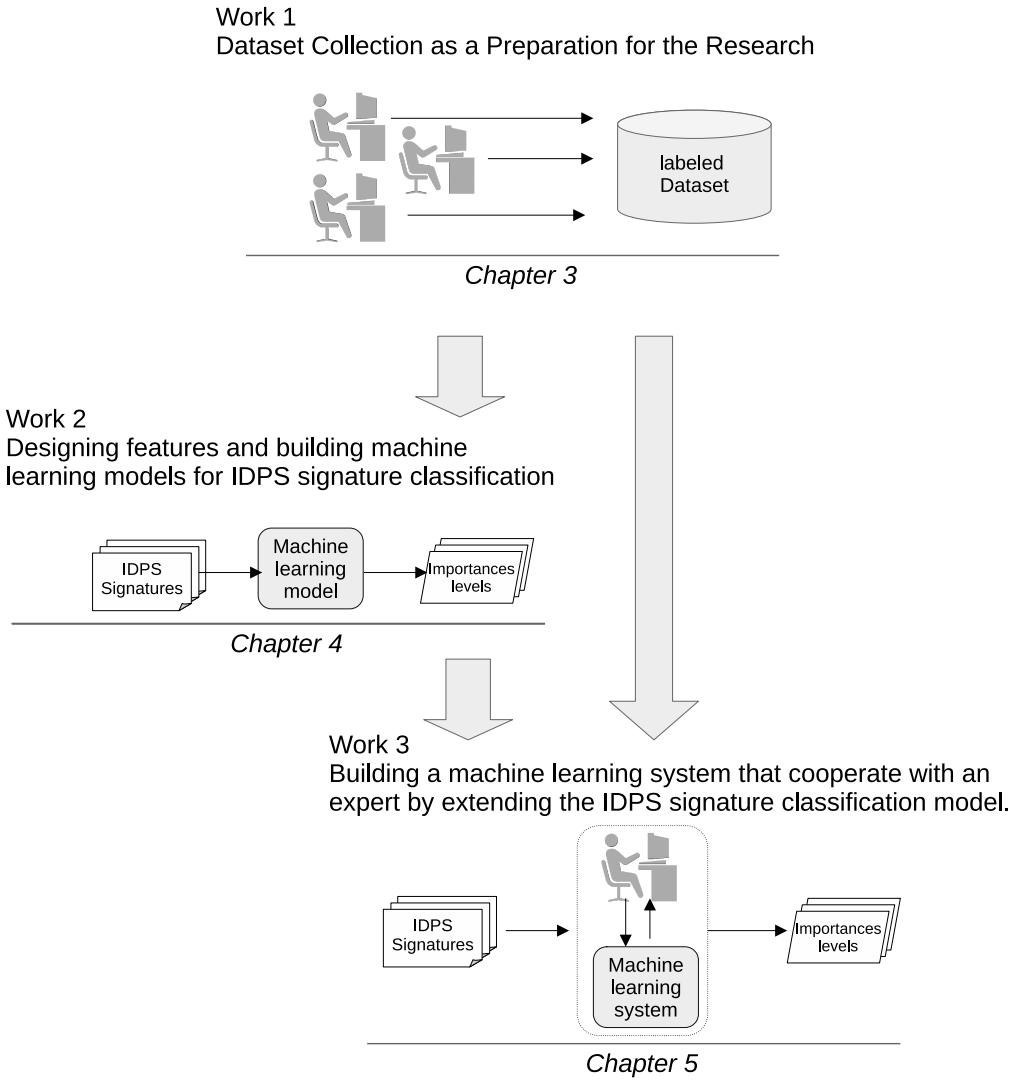


Figure 1.3: A roadmap of this research

# Chapter 2

# Related works

This research is related to the following two research areas.

- IDPS research area
  - Improvement of IDPS detection performance (Section 2.1.1)
  - Reducing the costs of IDPS operations (Section 2.1.2)
- Machine learning research area
  - Reject option (RO) (Section 2.2.1)
  - Active learning (Section 2.2.2)
  - Uncertainty estimation in deep learning (Section 2.2.3)

An overview of the relationship between these research areas and this research is shown in Figure 2.1. There are two types of research on IDPS: research to improve the performance of IDPS in detecting malicious communications and research to reduce the operational cost of IDPS. This research belongs to the latter category and uses machine learning to reduce the cost of signature management. Among the many areas of machine learning, ROs, active learning, and uncertainty estimation in deep learning are relevant. RO and active learning are necessary to realize the behavior for cooperation between humans and systems. Uncertainty estimation methods are introduced in the proposed system to identify signatures at high risk of misclassification accurately. In the following sections, we describe these research areas in detail.

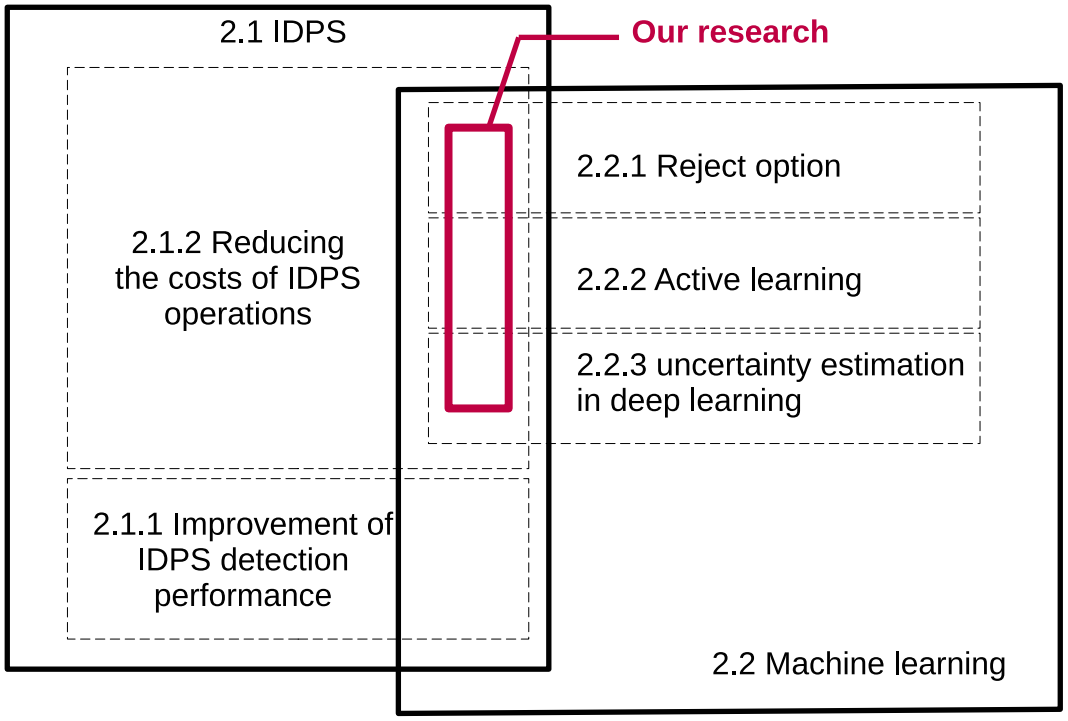


Figure 2.1: This reseaarch and related works

## 2.1 Intrusion detection and prevention system (IDPS) research area

### 2.1.1 Improvement of IDPS detection performance

IDPS performs logging, notification, blocking, and other actions on communications that match predefined malicious communications. There are two ways to express malicious communication patterns: signatures and machine learning classification models. IDPS vendors manually design and provide signatures by analyzing cyber attacks. Machine learning-based classification models are also generally developed by vendors.

The signature design is costly for vendors. In order to reduce this cost, research has been conducted to generate signatures automatically. Shahriar and Bond have proposed a method for automatically generating new signatures using genetic algorithms from existing signatures [20]. Others have applied decision trees (DTs) to generate signatures, such as Fallahi et al. automatically and Lee et al. have applied the Latent Dirichlet Allocation (LDA) to generate signatures automatically [21, 22].

There have been many studies on machine learning models for classifying nor-

mal and malicious communications [2]. In this case, the machine learning classification model takes the features of the malicious communication as input and outputs a multi-class classification of whether the communication is malicious or not or the type of malicious communication. Support Vector Machines (SVMs), DTs, bagging, artificial neural networks, and other methods have been applied [23, 24, 25, 26]. Unified benchmark tests (NSL-KDD, UNSW-NB15, and TUIDS) have been established for research in this field, and many research results have been reported [27, 28, 29].

These studies contribute to improving the performance of IDPS in detecting malicious communications. This research does not belong to the above studies. However, research to reduce the operational cost of IDPS is also being conducted, and this research belongs to that field.

### 2.1.2 Reducing the costs of IDPS operations

While it is important to improve the detection performance of IDPS, it is also important to operate IDPS efficiently in network security operations. The most burdensome part of IDPS operation is responding to alerts caused by false positives and managing signatures.

Usually, many alerts are due to false positives in the IDPS, and users of the IDPS are forced to deal with these alerts daily. Research is being conducted to analyze alerts from IDPSs and reduce the number of alerts themselves. Tadeusz proposes a system that incorporates machine learning to reduce false alerts [30]. Alsubhi et al. propose a fuzzy theoretical system to estimate the priority of alerts [31]. Cortés and Gómez propose a strategy that integrates several excessive alert reduction methods [32]. There are several other studies to reduce false alerts [33, 34].

Our approach differs from any of these methods in that it contributes to reducing the overall IDPS management costs by reducing the cost of setting up an IDPS. Research has been done to organize the signatures that are being created every day properly. Stakhanova et al. propose an analytical model for finding conflicting signatures [35]. In their model, signatures are represented as nondeterministic automata, and signature overlap is detected based on automata equivalence. Massicotte and Labiche propose another approach based on set and automata theories for the same purpose [36]. Other research efforts include identifying duplicate signatures [35, 36, 37] and methods for defining patterns of normal communication and identifying signatures that match these patterns, i.e., misjudged signatures [38]. Our approach differs from any of these methods in that it contributes to reducing the overall IDPS management costs by reducing the cost of setting up an IDPS.

Our research belongs to the field of signature management. As described in

Chapter 1, the importance level of each signature must be determined for each case. However, to our knowledge, no research has been done to automate this determination. This is probably due to the sensitivity of the data being handled and the difficulty in creating datasets. In order to automate the determination of importance levels for signatures, it is necessary to model the tacit knowledge appropriately and the thought patterns of experts. In this dissertation, we propose a machine learning system with features based on experts' knowledge as an approach to enable such modeling. In Chapter 4, the proposed features are evaluated to identify points of interest for experts to focus on when performing signature classification. In Chapter 5, simulation evaluation of signature classification is performed in real-world time-series order. These evaluations are performed on three real datasets we collected with experts in Chapter 3. To our knowledge, these datasets have yet to be collected.

## 2.2 Machine learning research area

### 2.2.1 Reject option (RO)

The RO is a function that determines whether to cancel the classification itself and is pioneered by Chow [39, 40]. Figure 2.2 shows an overview of RO. It makes a threshold decision based on the confidence/uncertainty score of the prediction that the classification model estimates. The idea of dropping a classification according to certain criteria is widely used, and these studies are reviewed.

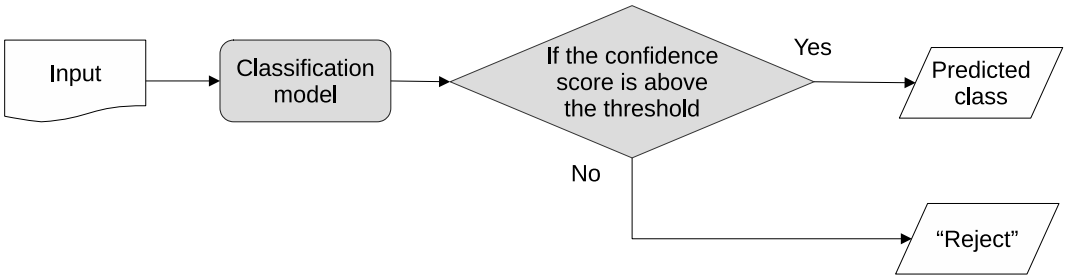


Figure 2.2: Reject option (RO)

Hanczar et al. proposed a method combining SVM and one-class SVM to improve the RO performance. Harish et al. performed a theoretical analysis of the RO in the case of three or more classifications [41]. Goepfert et al. extended the self-adjusting memory architecture (SAN-KNN) [42] and adaptive random forest (ARF) [43] methods to incorporate an RO for classification in nonstationary environments [44].

The RO is practical and has been studied for certain applications, especially in the medical field. Waseem et al. constructed and analyzed a classification model with an RO that predicts cancer based on genetic information [45]. Lin et al. proposed a classification model with an RO to classify biomedical images [46]. Their model uses SVM, which is an ingenious way to calculate a confidence score. The confidence score is based on the average ratio of the distance to the separating hyperplane during classification in the SVM, normalized to $[0, 1]$, and the distance to the centroid of each class in the feature space. Raghu et al. showed theoretically and experimentally that for images that are automatically classified by a classification model, the uncertainty of medical images can be directly estimated to determine whether to seek a second opinion [47]. An RO is helpful in fields where the impact of misclassification is significant, such as the medical field.

The classification model with the features proposed in Chapter 4 also measures RO performance. Several evaluation methods for RO have been proposed, with the accuracy-rejection curve (ARC) [48] being one of the most representative. The ARC is a visualization method of RO performance in which the trade-off between rejection rate and accuracy for a given test data (Figure 2.3). Note that accuracy in ARC is not the Top-1 classification accuracy (the simple percentage of correct answers to a classification problem) but rather the percentage of correct answers that are assumed to be correct even if rejected. The accuracy and rejection rates are computed for each threshold value used to determine the classification cancellation. The area under the ARC (AU-ARC) allows RO performance comparisons regardless of the threshold. Accuracy rejection normalized-cost curves (ARNCCs), an ARC-extended method, has also been proposed by Abbas et al. [49]. ARNCCs allow ARCs to take into account the cost of misclassification. Condessa et al. also proposed three RO performance metrics: nonrejected accuracy, classification quality, and rejection quality [50].

In IDPS signature classification, the subject of this research, the risk of failure is as high as in medical fields. This is because malicious communications may be missed due to misconfiguration of the IDPS and lead to security incidents. Security incidents should be avoided because they damage public trust in the organization operating the network. If regular communication is interrupted, the convenience of the network is also reduced. To mitigate such risks, using an RO in the classification model of IDPS signatures is a natural solution.

### 2.2.2 Active learning

Active learning is a framework that aims to complete the training task with minimal annotation costs [16, 51]. In active learning, predictive models are trained by selecting the samples from an unlabeled dataset that are most likely to be useful for training and having an annotator (also called an oracle) label them while building

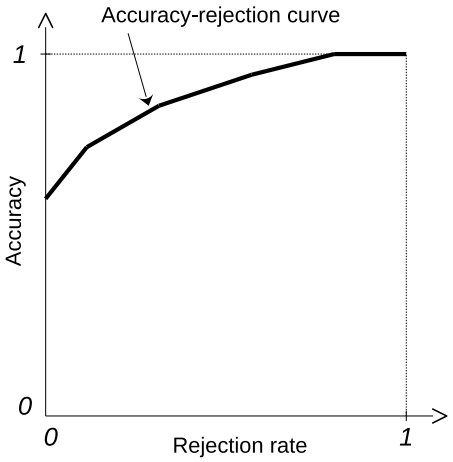


Figure 2.3: Accuracy-rejection curve (ARC)

the training data. Figure 2.4 shows an overview of active learning. First, in active learning, a machine learning model is trained from a training dataset containing a small number of labeled examples. Next, an acquisition function is used to select examples that are expected to facilitate better training the machine learning model from the unlabeled dataset. The oracle annotates the selected examples and adds them to the training dataset. Again, the machine learning model is trained on the updated training dataset. Active learning repeats the above process.

Labeling is costly in fields that require expertise, so researched it is being actively conducted to overcome this challenge. Active learning is applied to medical image processing [52, 53, 54, 55, 56], clinical text classification [57, 58], machine translation [59, 60, 61, 62, 63], chemical scenarios [64, 65, 66], and patent classification [67].

Our proposed system in Chapter 5 is a natural integration of this active learning paradigm into an expert's periodic signature classification task. This means that the cost of annotation is reduced for experts. The most popular main fields of active learning research are image and NLP. Other data types are relatively less explored, and to the best of our knowledge, there are no examples of applications of active learning to signature data structures.

### 2.2.3 Uncertainty estimation in deep learning

When using uncertainty sampling [16] as the acquisition function for active learning, the accuracy of the estimated probabilities output by the classification model is important. If the accuracy of the estimated probabilities is low, the uncertainty cannot be properly estimated, and better training samples cannot be selected.

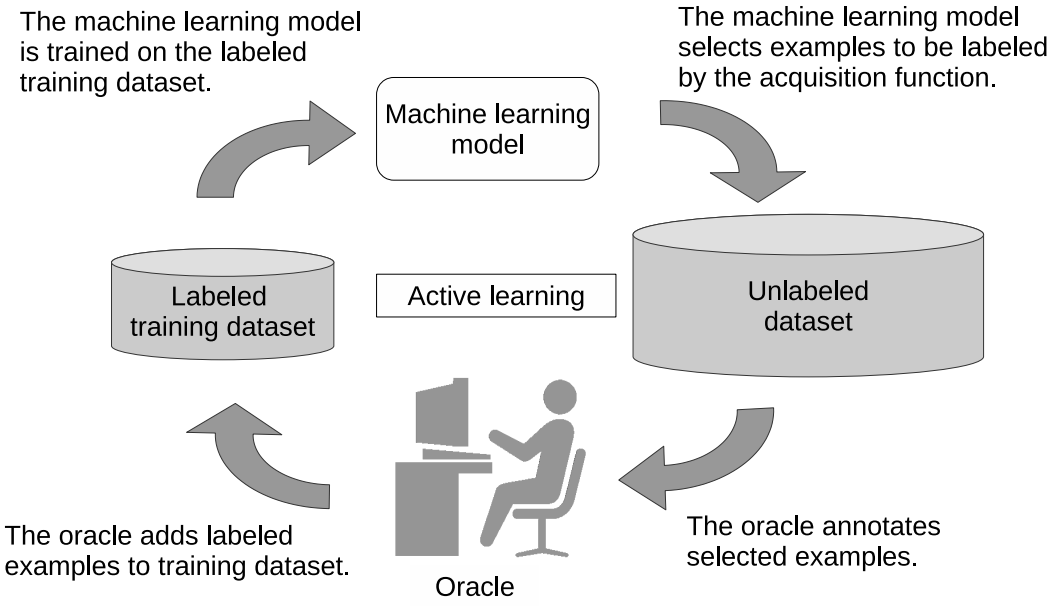


Figure 2.4: Active learning

An interesting topic in deep learning that has recently been applied in many fields is the calibration of DNNs, which is the process of correcting the predicted probabilities estimated by a DNN to the actual probabilities [17]. This is expected to be synergistic with uncertainty sampling, which requires the proper estimation probabilities to be determined. Since Guo et al. reported that large DNNs tend to be overconfident, further research has been conducted [68]. Calibration approaches include post hoc methods [69, 68, 70, 71, 72, 73, 74, 75, 76] that modify estimations after the fact and regularization methods [77, 78, 22, 79, 80, 81, 82, 83, 84] that modify the objective function and augment the data. In cascaded inference systems (a system in which a smaller DNN first performs inference, and if the result is uncertain, the decision is left to the larger DNN), calibration is crucial and discussed to optimize the trade-off between classification accuracy and computational cost [85].

Because of the difficulty of observing the actual probability values, evaluating calibration methods is not easy. Therefore, evaluation methods are also discussed. The most popular calibration metric is expected calibration error (ECE) [86], and many extensions have been proposed [87, 72, 88, 74], such as classwise-ECE [71] and adaptive calibration error (ACE) [89]. In addition, negative log-likelihood (NLL) [90], a popular measure for uncertainty estimation, is often used to evaluate the effects of calibration because it indirectly represents a calibration evaluation [68, 81, 76].

Basic research on calibration methods has been mainly evaluated using image classification benchmarks, e.g., CIFAR10[91], CIFAR100[92], SVHN[93], and ImageNet[94]. However, since accurately measuring the reliability of DNN predictions is also valuable for translation and dialogue systems, several experimental validations have been conducted in NLP [95, 83, 96, 88, 97]. These verifications use bidirectional encoder representations from transformers (BERT) [10]. BERT achieved state-of-art at the time for many natural language tasks, so its impact was significant, and calibration was likely taken up as a new research perspective based on BERT.

Many methods have been proposed to express the uncertainty of DNNs, and these techniques are also effective for calibration. These are the mainstream Bayesian methods [98, 99, 100, 101, 102, 103, 104, 18, 105], but the power of DE as non-Bayesian approaches have also been demonstrated [19]. The DE is a simple method that trains multiple DNNs and uses the average of these outputs as the final output when making predictions (Figure 2.5). DEs have the disadvantage of using multiple DNNs, which requires an enormous computational cost for even one. However, the calibration performance of DE has been empirically shown to exceed that of post hoc methods and Bayesian neural networks [106, 107, 108]. Increasing diversity among DNN members is said to improve calibration performance, and methods have been proposed for this purpose [109, 110]. Ashukaha et al. proposed an evaluation metric called deep ensembles equivalent (DEE) as a method to evaluate calibration performance based on DE. As a use case for DE, Jiang et al. proved that, given the assumption that DEs are calibrated, the degree of disagreement between DE members' predictions is expected to match the test error rate [111]. Thus, DEs have the potential to be used in a variety of ways, not only to improve calibration performance.

Another uncertainty estimation method for the DE, MC-Dropout [18], is popular due to its ease of implementation. MC-Dropout is an approximation method for Bayesian neural networks, in which dropout [112], which is usually used to reduce overfitting of DNNs, is also used to predict test data, and the average of multiple feed-forward outputs is used as the final output (Figure 2.6). A variation of MC-Dropout that performs dropout at only some layers has also been proposed and analyzed [113]. This dropout method at only some layers has been implemented on FPGAs to speed up the process [113].

Such methods also have calibration capabilities and are expected to be highly compatible with uncertainty sampling since the starting point better represents uncertainty. Our proposed system in Chapter 5 is based on active learning and uses uncertainty sampling as the acquisition function. This dissertation also verifies the performance of the proposed system by incorporating uncertainty estimation methods, such as the DE and MC-Dropout. The DE is introduced as a

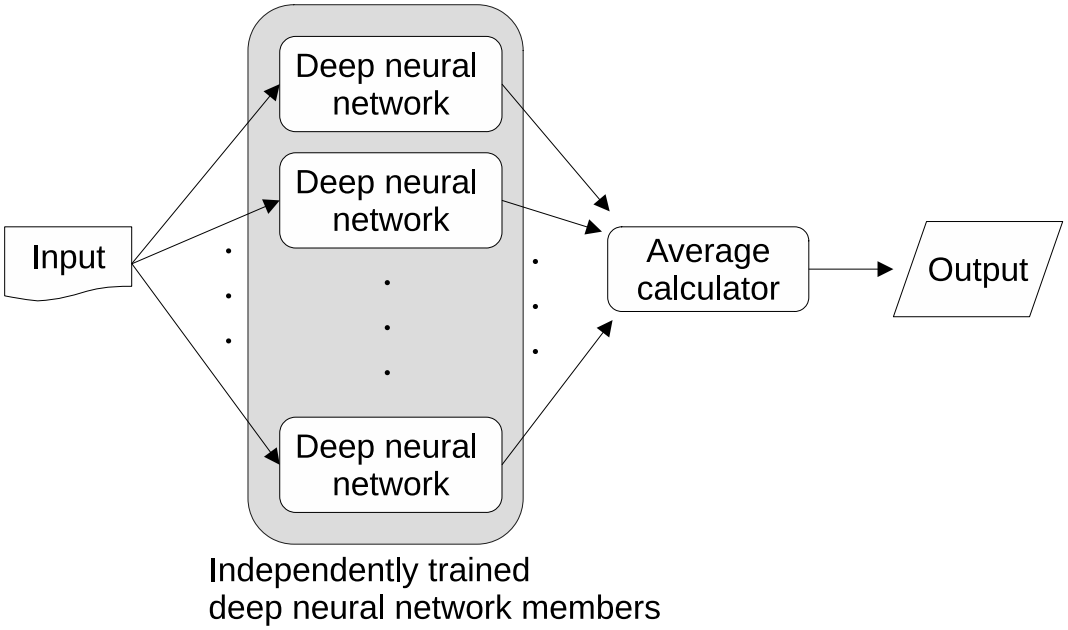


Figure 2.5: Deep ensembles (DE)

machine learning algorithm into the proposed model and the proposed system in the experiments in Chapter 4 and Chapter 5. MC-Dropout is incorporated into the proposed system in Chapter 5. These approaches have become particularly representative among uncertainty estimation methods, partly due to their ease of implementation, and further work on such methods is still being conducted today. In particular, [114] reported that they could be made better by combining active learning and DE.

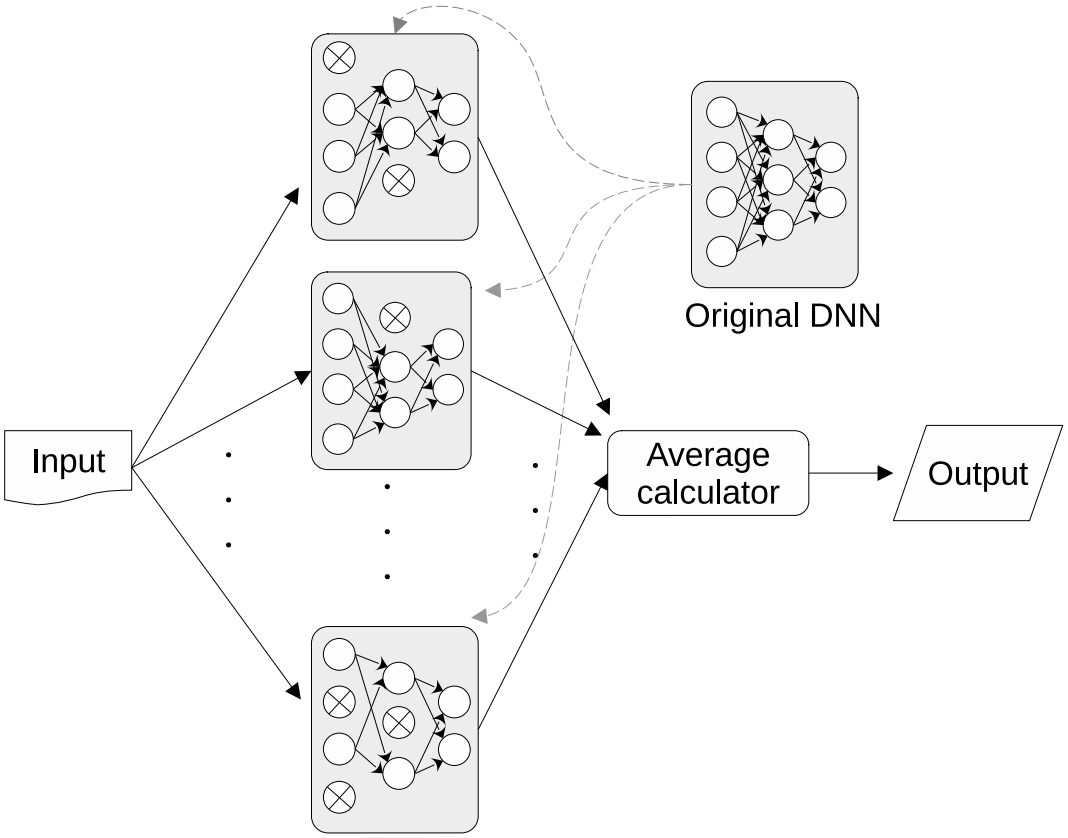


Figure 2.6: Monte Carlo dropout (MC-Dropout)

# Chapter 3

# Tasks and Datasets

## 3.1 IDPS signature classification procedure by experts

This research concerns IDPS signature classification for an expert in a real network operations organization. This chapter describes the assumptions of this research based on interviews with experts.

Experts classify the signatures distributed periodically by IDPS developers, like subscription services, based on their levels of importance. There are three levels of importance: “low”, “medium”, and “high”. These importance levels correspond to the action of IDPS when the signature is matched. If it is “low”, it neither notifies the expert nor intercepts the communication. In the case of “medium”, only notification is made. In the case of “high”, the communication is intercepted in addition to the notification.

Signatures are labeled by importance, but all classes are treated equally in this research. Suppose that “high” importance corresponds to the IDPS setting “block” and that “low” importance corresponds to “logging”. Classifying a signature labeled “high” as “low” would allow communications that should be blocked to pass through. Naturally, experts would prefer not to do this, as it could cause a security incident. Similarly, mistaking “low” for “high” should also be avoided. This mistake can block communications that should otherwise be allowed to pass through unimpeded. This prevents the network from providing proper communications to its users. Which of the two error types is more important is determined by the operational policy of the experts. Our research is conducted from the standpoint of not emphasizing any particular policy.

The expert semi-automatically classifies signatures according to the following procedure. The procedure is also shown in Figure 3.1. First, classification is performed by applying an if-then rule designed by them. The if-then rule assigns an

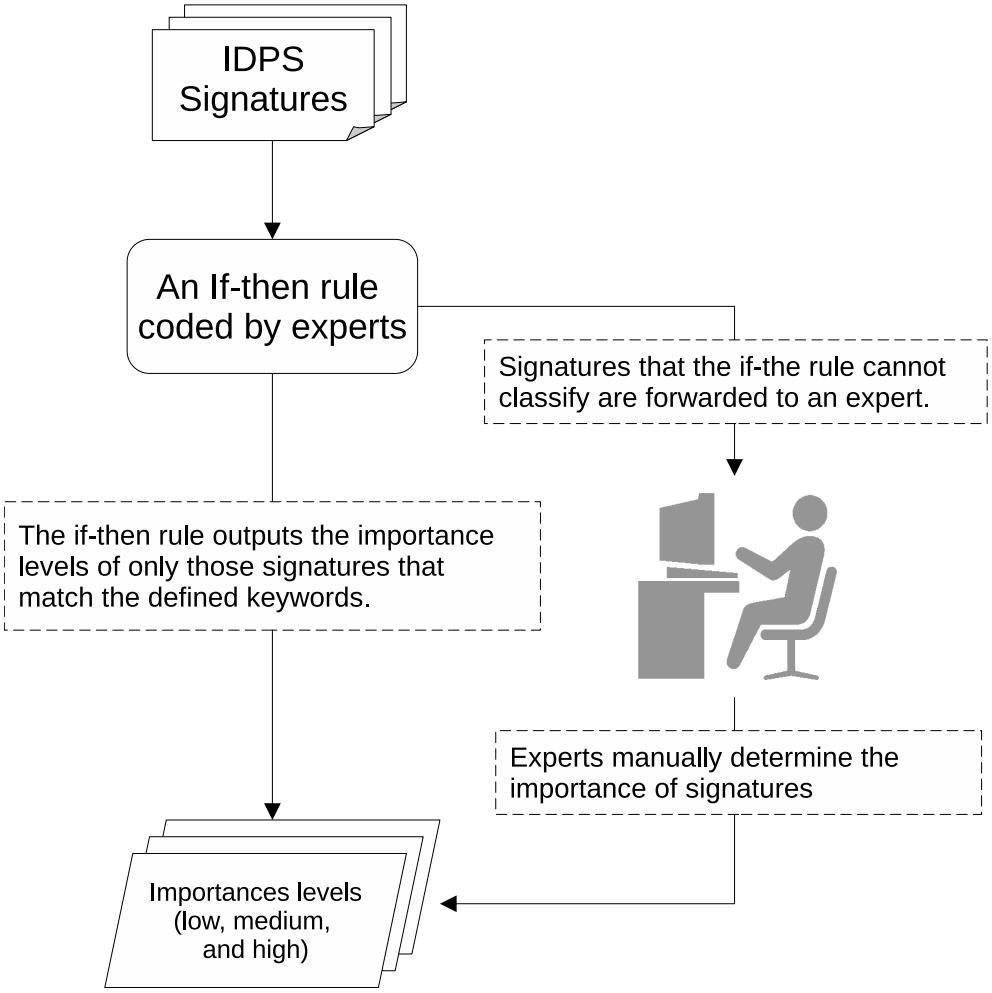


Figure 3.1: Signature classification procedure by experts

importance label or a label meaning "unknown" to a signature based on a keyword-matching combination of elements in the signature. Signatures determined to be unclassifiable by the if-then rule are then manually classified.

Since experts are motivated to classify signatures with if-then rules whenever possible, if-then rules are regarded as all the explicit knowledge that experts can explain. In contrast, the manual classification of signatures by experts is based on tacit knowledge. According to the experts, approximately 80 percent of the signatures can be classified using if-then rules, but a great deal of effort is required to classify the remaining 20 percent. In this research, we aim to reproduce the decision-making process of experts in manual classification using machine learning.

## 3.2 Notation of signatures

The signatures are written in the notation of the IDPS engine called Snort [1]. Figure 3.2 shows a concrete example[2]. The first word "alert" is the action taken by the IDPS when the signature is matched. Because the experts set up actions based on an importance level, actions cannot be entered into the classification model. Features are extracted from the strings after the action.

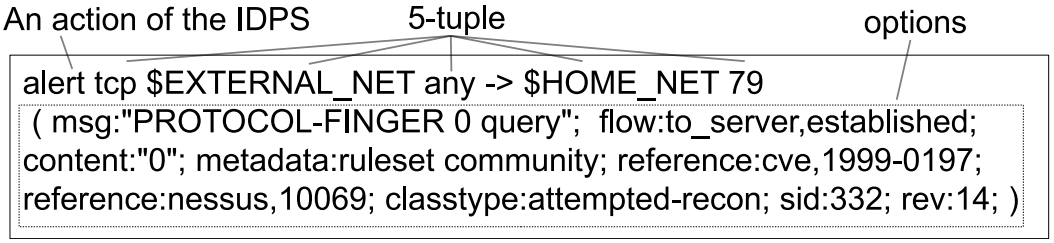


Figure 3.2: A specific example of IDPS signatures

"tcp $EXTERNAL_NET any -> $HOME_NET 79" is a 5-tuple of values. The 5-tuple is a set of five values listed in the header of an IP packet. "tcp" is the communication protocol. "$EXTERNAL_NET" is the source IP address. "any" is the source port number. "$HOME_NET" is the destination IP address. "79" is the destination IP address. A 5-tuple is an essential element of signatures. A string in parentheses after the 5-tuple is optional. The options are expressed in key-value format with the following conditions:

- The key and value are linked with colons.
- Semicolons are used as key-value separators.
- Depending on the key, there may be more than one value.
- Some values do not have a key, such as *nocase*.

We focus on four elements in the options: *msg* (abbreviation for message), *metadata*, *reference*, and *classtype*. The if-then rule classifies signatures in terms of 5-tuples and these elements alone.

*msg* is a string written to a log or alert when a signature is matched with a communication. "PROTOCOL-FINGER 0 query" in Figure 3.2 is an example of this.

[1] `https://snort.org/`

[2] `https://www.snort.org/downloads/#rule-downloads`

*metadata* is an element that represents information in the key-value format. The "ruleset community" in Figure 3.2 is an example of *metadata*. A space separates the key and value. Commas delimit key-value sets. This example has one key-value set.

*reference* describes a pointer to external information about the attack identification system. In Figure 3.2, it is described as "cve,1999-0197". In this example, *reference* refers to common vulnerabilities and exposures (CVE) 1999-0197. There are two ways to describe a *reference*. The first is the name and ID of the vulnerability list. In addition to nessus, there are CVE and Bugtraq. A CVE with ID 1999-0067 is listed as "cve,CVE-1999-0067", and Bugtraq with ID 629 is listed as "bugtraq,629". Second, the URL is directly described as the destination for accessing the information. For example, "`url,www.spywareguide.com/product_show.php?id=973`".

*classtype* is a general group of malicious communications indicated by signatures. In Figure 3.2, it is "attempted-recon". The groups that *classtype* indicates are different from the importance levels determined by the experts.

## 3.3 If-then rule for IDPS signature classification

The experts coded an if-then rule script to classify as many signatures as possible. An if-then rule returns a label of "low", "medium", "high", or "unknown" importance based on keyword matching of the elements in the signature.

The if-then rule classifies signatures by matching keywords and combinations of keywords. Keyword matching is used to determine whether a word is included in a signature. The key-value pairs used in keyword matching are the 5-tuple, *msg*, *metadata*, *reference*, and *classtype*. Keyword matching for *metadata* uses a key-value pair as one keyword. *msg* keyword matching does not consider word position. In other words, keyword matching for *msg* determines whether a certain word appears. Keyword matching for *reference* determines whether a specific system is referred to, and it does not use an ID. The elements of the 5-tuple are extracted and judged individually. Because *classtype* is represented by a single symbol, no special preprocessing is applied.

The number of keywords extracted for the if-then rule is 133 for 5-tuple, 2 for metadata, 56 for msg, 1 for reference, and 6 for *classtype*. The if-then rule contains 61 conditions that combine logical products and logical sums, with keyword matching as the basic component. The number of conditions is smaller than the number of keywords because determinations also require matching with multiple keywords. As specific examples of keywords, conditions with "high" importance include "trojan-activity" in *classtype* and "MALWARE-TOOLS" in *msg*. For "medium" importance, there are "network-scan" in *classtype*, "blacklist" in

*msg*, and so on. If the importance is "low", the source IP address of the 5-tuple is "$EXTERNAL_NET", the source IP address of msg is "MALWARE-CNC", etc. If the importance is "moderate", the source IP address of msg is "MALWARE-CNC", etc.

The if-then rule assigns importance labels only to signatures that match the keyword match condition. Signatures that do not meet the conditions are assigned an "unknown" label. The conditions are prioritized so that if a signature matches more than one condition, the importance label for the condition with the highest priority is assigned.

## 3.4 Collecting labeled IDPS signature datasets

In this research, the target dataset is the signatures labeled by experts in actual network security operation organizations to be set in the IDPS for their actual work. The labels for each signature were determined through consultation among several experts for this research. Different datasets are used for the evaluation of the feature design and the evaluation of the proposed active learning-based system. Below are the names of the databases used in each chapter.

**Chapter 4**

- Automatically annotated dataset (AAD)
- Manually annotated dataset (MAD)

**Chapter 5**

- Time-series manually annotated dataset (TMAD)

The collection process and details of each dataset are shown below.

### 3.4.1 Automatically annotated dataset (AAD) and manually annotated dataset (MAD)

First, the experts automatically classify signatures with the if-then rules. The experts coded if-then rules so that as many importance labels as possible could be assigned. Next, the experts manually determine the labels of signatures that do not match the if-then rules. Two datasets are then created: one for signatures classified by the if-then rule and the other for signatures classified manually. The former is called the AAD, and the latter is called the MAD.

Table 3.1 shows the number of AAD and MAD samples prepared by the experts. Each signature is assigned one of three importance labels: low, medium, or

Table 3.1: Summary of AAD and MAD

| Dataset | Priority | | | Total |
|---|---|---|---|---|
| | low | medium | high | |
| AAD | 3,936 | 93 | 436 | 4,465 |
| MAD | 1,119 | 122 | 59 | 1,300 |

high. Based on the importance level, the experts set the action of the IDPS for communications that match the signature.

IDPSs are operated to monitor the communication of servers that actually provide services on the Internet. This service is available only to users who have contracted with an organization affiliated with the expert. The datasets in this research (AAD and MAD ) are real datasets created for use in IDPS. The signatures that make up the dataset were distributed by the company that develops and sells the IDPS for approximately one year and six months, from December 2016 to May 2018.

For practical use, there is no need to use machine learning models to classify AADs, which can be classified by if-then rules. However, since the purpose of this research is to construct a signature classification model, experiments are also conducted on AADs to confirm the degree to which the proposed features and machine learning can simulate the if-then rule.

### 3.4.2 Time-series manually annotated dataset (TMAD)

To evaluate the proposed system in Chapter 5, we develop a real dataset consisting of time-stamped and labeled signatures with the help of experts. In this experiment, we only collected signatures that could not be classified by the if-then rule, i.e., those that require manual classification by an expert.

Table 3.2 shows the distribution of the classes by time step $t$. Signatures are classified into one of three importance levels, low, medium, or high, and assigned a time step $t$. The signatures and their labels were collected on a monthly basis for two years. These signatures are distributed periodically by an IDPS developer. Some signatures can be automatically classified using if-then rules, but these signatures are thinned out in advance. In other words, the dataset consists only of signatures that experts have manually labeled based on their knowledge and experience.

Table 3.2: Summary of TMAD

| Time step | #. low | #. medium | #. high | total |
|---|---|---|---|---|
| $t = 0$ | 155 | 16 | 3 | 174 |
| $t = 1$ | 281 | 1 | 1 | 283 |
| $t = 2$ | 232 | 16 | 9 | 257 |
| $t = 3$ | 566 | 36 | 3 | 605 |
| $t = 4$ | 214 | 18 | 2 | 234 |
| $t = 5$ | 643 | 20 | 4 | 667 |
| $t = 6$ | 326 | 7 | 6 | 339 |
| $t = 7$ | 364 | 8 | 1 | 373 |
| $t = 8$ | 516 | 6 | 3 | 525 |
| $t = 9$ | 219 | 26 | 4 | 249 |
| $t = 10$ | 218 | 15 | 4 | 237 |
| $t = 11$ | 357 | 6 | 2 | 365 |
| $t = 12$ | 173 | 0 | 5 | 178 |
| $t = 13$ | 285 | 20 | 14 | 319 |
| $t = 14$ | 203 | 19 | 2 | 224 |
| $t = 15$ | 217 | 43 | 7 | 267 |
| $t = 16$ | 291 | 43 | 3 | 337 |
| $t = 17$ | 290 | 74 | 7 | 371 |
| $t = 18$ | 165 | 36 | 5 | 206 |
| $t = 19$ | 314 | 35 | 1 | 350 |
| $t = 20$ | 214 | 49 | 6 | 269 |
| $t = 21$ | 174 | 44 | 1 | 219 |
| $t = 22$ | 276 | 64 | 0 | 340 |
| $t = 23$ | 143 | 39 | 7 | 189 |
| total | 6,836 | 641 | 100 | 7,577 |

## 3.5 Limitation

The importance level of a signature depends on the information and communication systems monitored by the IDPS. In other words, even if the same expert determines the importance level of the same signature, the importance level of the signature may differ depending on the network systems monitored by the IDPS. Therefore, the IDPS sigclassification model/system should be constructed for each information and communication system monitored by the IDPS.

The construction and evaluation of signature classification models for multiple information and communication systems and the analysis of differences in classification models among information and communication systems are future work and are not addressed in this research. We cannot guarantee the same accuracy when using datasets collected by other information and communication systems. However, the methods used by the experts to create datasets, extract features, and construct classification models are independent of the information and communication systems monitored by the IDPS. In order to demonstrate the generality of the findings in this dissertation, it is necessary to collect similar datasets and perform similar validations in multiple organizations. However, this research contributes to the demonstration of its generality.

# Chapter 4

# Feature engineering based on experts' knowledge

## 4.1 Problem setup

To classify IDPS signatures by machine learning, it is necessary to search for an effective feature extraction method. In this chapter, we design and experiment with features based on the expert signature classification procedure, AAD and MAD introduced in Chapter 3.

Because the signatures are written in text, they can be fed into large-scale language models [10, 11, 12, 13, 14]. However, the predictions obtained from these neural network-based models are difficult to interpret. In other words, even if the classification model performs well, we do not know why. For a first step, it is desirable to make interpretation easy, so that future research can be developed. Our policy is to design features that are easy to interpret while still using machine learning. Specifically, we design features so that it is possible to identify which elements of the signature are focused on by experts.

We classify all signatures with a single classification model in order to analyze the experts' decision process in later evaluation experiments (Chapter 4.3). We design and represent the signatures as a common feature vector regardless of whether the signatures conform to the if-then rule. Referring to the classification procedures of the above experts, we designed three feature vectors:

1. Features that are subject to the conditions of the if-then rule.
2. Features obtained from keywords in the if-then rule.
3. Features obtained by web scraping from messages and external reference information in the signatures.

## 4.2 Proposed features

To construct a classification model, we propose *SFs*, *KFs*, and *WMFs*. The procedure for extracting these features and their relationship is shown in Figure 4.1. The SFs and KFs are designed with reference to if-then rules. The WMFs are designed with reference to interviews with experts. SF is extracted from *5-tuple*, *metadata*, and *classtype*, while KF and WMF are extracted from *msg* and *reference*.

### 4.2.1 Symbolic features (SFs)

SFs are extracted from the *5-tuple*, *metadata*, and *classtype*, each of which is extracted as a feature with one-hot encoding. The processing procedure is shown in the gray box on the left side of Figure 4.1. The one-hot encoding is a method of converting nominal features into a numeric vector. For example, if there are three kinds of symbols, A, B, and C, they are converted to features $[1, 0, 0][0, 1, 0]$, $[0, 0, 1]$ respectively.

The *classtype* is directly extracted as a feature using one-hot encoding. However, the 5-tuple and *metadata* need to be preprocessed. The 5-tuple is separated into its five values. After that, each value is converted into features by one-hot encoding. In the extraction procedure for *metadata*, all the key-value pairs are extracted first. Then, all the extracted key-value pairs are reordered and combined into a string to form a single symbol. This sorting eliminates the influence of the order in which key-values appear in the metadata. In principle, there is a huge variety of key-value combinations, but the number of such combinations that appear in AAD and MAD is small.

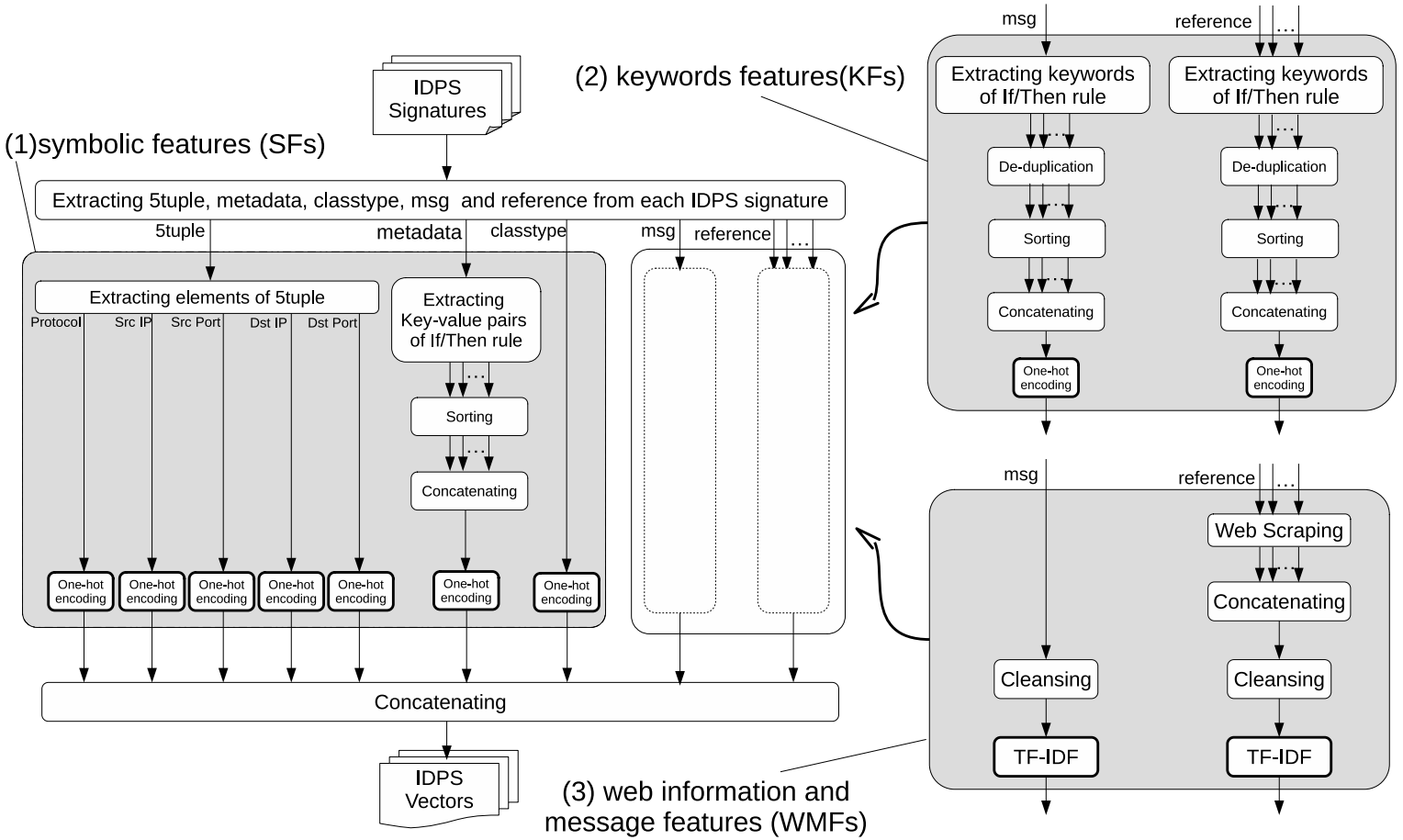


Figure 4.1: Proposed features for signature classification: (1) *symbolic features*, (2) *keyword features* and (3) *web information and message features*.

### 4.2.2 Keyword features (KFs)

KFs are designed for keyword matching on *msg* and *reference* in the if-then rule. The KF processing procedure is shown in the gray box in the upper right corner of Figure 4.1. KFs are extracted from *msg* and *reference* according to the presence or absence of the keywords in the if-then rule. After extraction, they are converted with one-hot encoding.

To convert *msg* to features, a list is created from the words used in the matching conditions from the if-then rule. Sets of words in the list are extracted as symbols from *msg* in the order of the list. If no word matches the word list, a dummy symbol is used to indicate that the word does not exist. Then, the symbols are converted with one-hot encoding.

To extract features from *reference*, a list is made from system names that exist in the if-then rule. The system names pointed to by *reference* are combined and treated as a symbol. If a system name does not match the list, it is extracted as a dummy symbol indicating this. These symbols are converted to features by one-hot encoding.

### 4.2.3 Web information and message features (WMFs)

Many signatures cannot be classified by an if-then rule. We need to add a new criterion to capture the properties of such signatures.

We interviewed experts to design new features from the criteria used by experts to classify features manually. The expert determines the importance level of a signature based mainly on the external information indicated by the *msg* and *reference*, the configuration information of the information and communication system to be operated, and his/her own experience and knowledge. Specifically, first, the expert understands the type and detailed characteristics of the malicious communication targeted by the signature from the external information indicated by the *msg* and *reference*. Then, the importance level is determined by considering the degree of the adverse impact of the types and characteristics on the information communication system and the risk of over-detection or false detection. For example, if the type of malicious communication is a SQL injection that attacks a database, and the database is managed in the information and communication system, the importance level is likely to be set to "medium" or "high". The results of the interviews are summarized in the following two insights.

- The experts check all the information in *msg*.
- The experts check the external information of *reference* via a web search.

We assume that the whole *msg* and the information on the web indicated by *reference* are essential and propose features that can effectively use them. To ex-

pand on this information, we apply the term frequency-inverse document frequency (TF-IDF), which is frequently used in natural language analysis [115, 116, 117]. Web scraping is also used for feature extraction on *reference*. The gray box at the bottom right of Figure 4.1 shows the procedure for extracting WMFs.

For *reference*, web scraping is performed to obtain information from external references. *reference* is a set of names of vulnerability lists (CVE, Bugtraq, etc.) and their IDs or URLs, which allow information related to the signature to be uniquely identified. For example, when referring to CVEs, information can be obtained by searching for an ID in web systems such as the National Vulnerability Database (NVD)[1] and RedHat's CVE Database[2]. Examples of signature-related information include the software targeted by the malicious communication indicated by the signature and its version information. The developer of the classification model needs to describe the web scraping process for each web system that publishes the information referred to by the *reference*. Although applying the procedure to all web systems is difficult, it is possible to describe the procedure by focusing on frequently used web systems. In what follows, *reference* refers to information obtained by web scraping.

Before extracting TF-IDF values, *msg* and *reference* are cleaned as follows: Since only alphabets, numbers, and underscores are used, other symbols are replaced by blanks. Stop words [118] and words that appear only once in all signatures are removed.

The cleaned *msg* and *reference* are converted into feature vectors by TF-IDF separately. Let $d$ be an identifier for a text (*msg* or *reference* in a signature) and $t$ be an identifier for a word; the TF-IDF is as follows:

$$tf\text{-}idf(t,d) = tf(t,d) \cdot idf(t) \tag{4.1}$$

$tf(t,d)$ represents the number of occurrences (an integer greater than or equal to 0) of the word $t$ in the text $d$. $idf(t)$ is calculated as follows:

$$idf(t) = \log \frac{N_L + 1}{df_L(t) + 1} + 1. \tag{4.2}$$

$N_L$ is the number of texts in the training data. $df_L(t)$ is the number of texts in which the word $t$ appears among the $N_L$ training samples. In other words, the IDF used when converting the test data to feature vectors with TF-IDF is the IDF calculated in the training data. When converting to TF-IDF, all words are treated as unigrams. After conversion to TF-IDF, L2 normalization is performed for each WMF. After L2 normalization, min-max scaling is performed with a minimum value of zero and a maximum value of one.

[1] `https://nvd.nist.gov/vuln`

[2] `https://access.redhat.com/security/security-updates/#/cve`

## 4.3 Evaluation of proposed features for machine learning models

Experiments are conducted on AAD and MAD to evaluate the performance of the proposed features. This experiment aims to check the validity of the proposed features and the focus of experts when performing signature classification. The time-series nature of the signatures is not considered.

### 4.3.1 Experimental setting

#### Outline of the experiment

In this section, we confirm the classification accuracy and RO performance of the classification model with proposed features and analyze the validity of the feature design. Specifically, evaluation experiments are conducted on the following process:

1. **Measuring classification accuracy**
   We validate the proposed features on several traditional machine learning models and compare the balanced accuracy (BACC) as a classification accuracy.

2. **Measuring of RO performance**
   We evaluate the quantified RO performance with ARCs and the AU-ARC [48].

3. **Measuring of DE for the RO**
   We explore the RO performance improvement by using a DE [19], which is said to better represent uncertainty.

4. **Analysis of the expert's point of view**
   We analyze the importance of the proposed features to identify the signature elements that the experts regard as important for evaluating the signature.

5. **Analysis of valid features**
   We explore which elements of the proposed features contributed to the classification accuracy.

The results of these experiments are shown in Section 4.3.2.

#### Extraction feature sets

We extract two types of feature sets. One concatenates the SFs and KFs into a vector directly. These connected features are called if-then rule features (ITRFs).

The ITRF is a feature design based on the if-then rule. Second, we directly concatenate the SFs and WMFs to form a vector. The connected features are called manual classification features (MCFs). The MCF is a feature design considering manual classification by experts.

### Implementation of web scraping

Before converting to WMFs, the information is extended by web scraping for each signature for both AAD and MAD. In this dissertation, we perform web scraping to obtain the software and version information of the target of the malicious communication indicated by the signature. We try to obtain the text indicating the target software and version information in the order of CVE, Bugtraq, and URL in the signature and terminate web scraping for the signature when the information is successfully obtained. When retrieving from CVE, search in the order of NVD and RedHat's CVE Database. When retrieving from Bugtraq, search from SecurityFocus[3]. When retrieving from a URL, attempt to retrieve if the URL refers to the Vulnerability Report[4] of Talosintelligence, Adobe Security Bulletin[5], or Exploit Database[6]. By the above procedure, we succeeded in obtaining information on $2,807$ out of $4,465$ for AAD and $1,024$ out of $1,300$ for MAD.

### Machine learning

In experiments using five different trained classification models, linear support vector machine (SVM), multilayer perceptron (MLP), decision tree (DT), random forest (RF), and naive Bayes (NB), we evaluate the ITRFs and MCFs. The numbers of samples of the two classes with low numbers are increased to the same level as that of the majority class by SMOTE [119]. The number of neighbors is 5. The hyperparameters for each machine learning model are shown below.

The classification model of linear SVM is trained with a regularization parameter $C = 1$. One-vs-rest (OvR), which can be applied to multiclassification problems, is used.

The MLP in this experiment consists of three layers with a hidden layer of 100 nodes and is trained by backpropagation. The activation function for all nodes is a rectified linear unit (ReLU; ramp function). Overfitting is suppressed by L2 regularization. The regularization parameter is set to 0.0001. We use adaptive moment estimation (Adam) [120] for the optimization of objective functions. The Adam parameters are set to the default values in [120] ($\alpha = 0.0001, \beta_1 = 0.9, \beta_2 =$

[3] `https://www.securityfocus.com/`
[4] `https://talosintelligence.com/vulnerability_reports`
[5] `https://helpx.adobe.com/security.html`
[6] `https://www.exploit-db.com/`

$0.99, \epsilon = 10^{-8}$). Training is terminated if the loss value in the training data is not less than 0.0001 in a minimum of 10 iterations.

The DT is trained using the classification and regression tree (CART) method with Gini impurity as the indicator. Training is performed until the number of samples or classes present at all endpoints reaches 1.

RF consists of 10 DTs trained the same way as DT. Each DT is trained by randomly selecting $\lfloor\sqrt{m}\rfloor$ features with $m$as the dimensionality of the feature vector. $\lfloor\cdot\rfloor$ represents the floor function, which is defined as the largest integer less than or equal to the input value.

NB is implemented by assuming that the input variables follow a normal distribution.

### Classification with an RO

In this experiment, RO performances also are evaluated using SVM and MLP among the classification models. The RO can be used on any classification model as long as a prediction score can be calculated. It is formulated as follows: Let $x \in \mathcal{X}$ be the input class and $y \in \mathcal{Y} = \{1, .., C\}$ be the output class. Let $S_y : \mathcal{X} \to \mathbb{R}$ be a function that computes the prediction score of class $y$ for a classifier. The final classification result $\hat{y}_i$ for the input $x_i$ of the classification model with RO is as follows.

$$\hat{y}_i = \begin{cases} \arg\max_{y \in \mathcal{Y}} S_y(x_i) & \text{if} \max_{y \in \mathcal{Y}} S_y(x_i) \geq \tau \\ \phi & \text{otherwise.} \end{cases} \tag{4.3}$$

$\tau$ is the threshold, which is the hyperparameter of the RO. $\phi$ is a symbol of rejection.

The following shows how the scores are calculated for each machine learning model. For the OvR linear SVM, the prediction score is the maximum distance from the decision boundary. MLP uses the maximum value of the prediction probability vector normalized by the softmax function as the prediction score.

### Evaluation method

Due to the imbalanced dataset, we measure the BACC as the classification accuracy. BACC is also called macro-Recall. Let $y$ be a symbol indicating a class. Let $\mathrm{TP}(y)$ be the number of samples that belong to the true class $y$ that were correctly predicted as $y$. Let $\mathrm{FN}(y)$ be the number of samples that belong to the true class $y$ that were incorrectly predicted to be in another class. The BACC is as follows.

$$\mathrm{BACC} = \frac{1}{C} \sum_{y=1}^{C} \frac{\mathrm{TP}(y)}{\mathrm{TP}(y) + \mathrm{FN}(y)}$$

BACC is the interclass average of the recall for a class $y$.

In addition, to verify the performance of the RO, we plot an ARC [48]. The ARC visualizes the trade-off between the classification accuracy and the rejection rate generated by the RO. Note that the classification accuracy is the top-1 classification accuracy, where the rejected samples are considered correct. Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a dataset containing $N$ labeled samples for evaluation. The following equations show the accuracy (vertical axis) and rejection rate (horizontal axis) of ARC at the threshold of the RO.

$$\begin{aligned} \text{Accuracy}(\tau) &= \frac{1}{N}\sum_{i=1}^{N} \min(\mathbb{1}(\hat{y}_i = y_i) + \mathbb{1}(\hat{y}_i = \phi), 1) \\ \text{Rejection rate}(\tau) &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = \phi) \end{aligned}$$

If the AU-ARC is included, it is possible to compare methods regardless of the threshold value. The experiments are performed with trained 10-fold cross-validation.

### 4.3.2 Experimental results

We perform a Stratified 10-fold cross-validation on all combinations of the two datasets (AAD and MAD), transformation methods to feature vectors (ITRF and MCF), and machine learning models (SVM, MLP, DT, RF, and NB). The mean and standard deviation of the dimensionality of the feature vectors are shown in Table 4.1. For each of AAD and MAD, the number of ITRF and MCF dimensions is listed for each of the five elements of the signatures focused on in this dissertation (*5-tuple*, *metadata*, *classtype*, *msg* and *reference*). The value on the left is the mean of the number of dimensions, and the value with ± in parentheses is the standard deviation. Each value is rounded to two decimal places.

#### Measuring classification accuracy

In each fold of the stratified 10-fold cross-validation, the test data were classified with the trained classification model. The BACC was then measured, and the mean and standard deviation were calculated across the 10 folds. Table 4.2 shows the results. The column for each machine learning model name indicates the value for BACC. The value on the left is the mean of the BACC, and the value in parentheses with ± represents the standard deviation. Each value is rounded to the fourth decimal place.

The experimental results for AAD show that ITRF performs sufficiently close to the if-then rule. On the other hand, ITRF shows a significant decrease in

Table 4.1: Dimension of the feature vector in the experiment

| Dataset | Features† | | #. Dimensions |
|---|---|---|---|
| AAD | ITRF | 5tuple | 236.8 (±3.0) |
| | | metadata | 427.3 (±4.8) |
| | | classtype | 15.7 (±0.7) |
| | | msg | 21.9 (±0.3) |
| | | reference | 2.0 (±0.0) |
| | MCF | 5tuple | 236.8 (±3.0) |
| | | metadata | 427.3 (±4.8) |
| | | classtype | 15.7 (±0.7) |
| | | msg | 1,760.8 (±10.3) |
| | | reference | 2,081.8 (±75.0) |
| MAD | ITRF | 5tuple | 96.8 (±1.8) |
| | | metadata | 383.1 (±6.2) |
| | | classtype | 9.9 (±0.3) |
| | | msg | 6.0 (±0.0) |
| | | reference | 2.0 (±0.0) |
| | MCF | 5tuple | 96.8 (±1.8) |
| | | metadata | 383.1 (±6.2) |
| | | classtype | 9.9 (±0.3) |
| | | msg | 803.8 (±9.1) |
| | | reference | 563.2 (±27.1) |

† ITRF is composed of the linkage of SF and KF, and MCF consists of the connection of SF and WMF.

accuracy for MAD compared to AAD. This indicates that MAD, a dataset that does not match the if-then rule, is difficult to classify with ITRF, composed of features designed concerning the if-then rule.

Next, to compare ITRF and MCF, we review the experimental results on MAD. We can confirm that MCF significantly outperforms all machine learning models. The performance of MCF is 30.43% for Linear-SVM, 27.47% for Multilayer Perceptron, 27.09% for DT, 25.13% for RF, and 38.94% for Naive Bayes, all of which show a minimum 0.251 improvement. MCF is used. The highest performance with MCF is 86.82% for Linear-SVM, and the lowest is 82.38% for RF. MCF achieves the lowest performance of 82.38% regardless of the machine learning model. The only difference between ITRF and MCF is whether KF or WMF is used to transform feature vectors for *msg* and *reference*. Therefore, WMF improved the accuracy by at least 25.13% for both machine learning models. These results suggest that WMF in MCF captures the characteristics of manual classification by experts well.

Table 4.2: Balanced accuracy between ITRFs and MCFs

| Dataset | Features† | SVM | MLP | DT | RF | NB |
|---|---|---|---|---|---|---|
| AAD | ITRF | 95.69 ±2.41 | 95.66 ±2.02 | 95.40 ±3.48 | **92.93 ±3.89** | 76.37 ±5.24 |
| | MCF | **96.86 ±2.63** | **96.43 ±2.42** | **96.27 ±2.98** | 92.65 ±3.09 | **88.26 ±5.24** |
| MAD | ITRF | 56.39 ±6.96 | 58.98 ±6.74 | 59.59 ±7.05 | 57.35 ±10.09 | 45.24 ±8.48 |
| | MCF | **86.82 ±6.35** | **86.45 ±6.99** | **86.68 ±6.96** | **82.38 ±4.42** | **84.18 ±7.57** |

† ITRF is composed of the linkage of SF and KF, and MCF consists of the connection of SF and WMF.

### Measuring of RO performance

In this section, we measure the performance of the RO for SVM and MLP. Table 4.3 shows the AU-ARCs. Figure 4.2 shows all the ARCs for each fold in the stratified 10-fold cross-validation. The overall trend is similar to that of BACC. On the AAD, there is no significant performance difference between the ITRFs and MCFs. However, on the MAD, the MCFs outperform the ITRFs for the RO. The RO performances of the linear SVM and MLP improve by 6.19% and 4.24%, respectively.

Table 4.3: AU-ARC (%) between the ITRFs and MCFs.

| Dataset | Features | SVM | MLP |
|---|---|---|---|
| AAD | ITRF | 99.93 ± 0.07 | 99.93 ± 0.06 |
| | MCF | **99.95 ± 0.05** | **99.98 ± 0.02** |
| MAD | ITRF | 93.26 ± 1.81 | 95.18 ± 0.95 |
| | MCF | **99.45 ± 0.36** | **99.42 ± 0.36** |

In real cases, experts classify signatures that are rejected by classification models. The RO performance on the MAD shows its effectiveness. On the MAD, the MCFs exhibit a high RO performance exceeding 99% AU-ARC when using the linear SVM and MLP. This result is one more indication of the practicality of the MCFs.

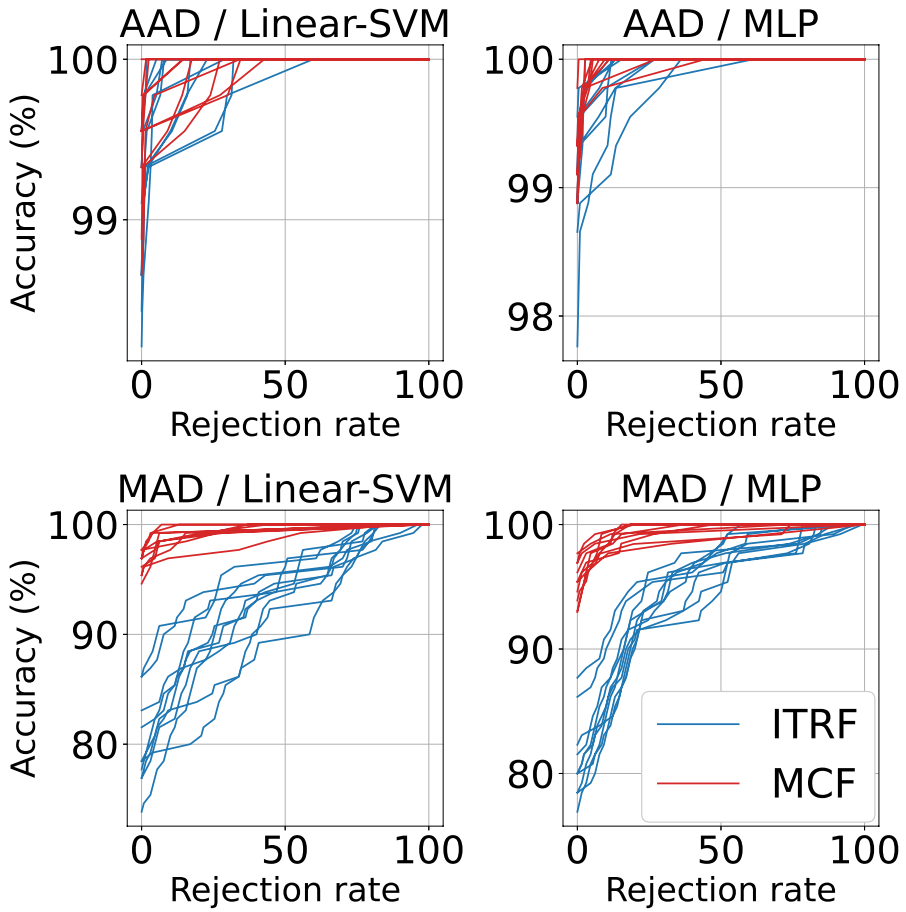


Figure 4.2: Each ARC shows the result of 1 fold in stratified 10-fold cross-validation.

### Measuring of DE for the RO

In this section, we use DE [19] as a classification model to further improve the performance of the RO and confirm its effectiveness. The RO is said to have a better trade-off between accuracy and rejection rate the closer its prediction score is to the actual probability that it fits the classification [39]. The DE is considered a method that better represents the uncertainty of DNNs. Experiments have also shown the superior calibration capability of the DE, which is the ability to estimate the true probability [19, 106].

The analysis results are shown in Table 4.4 and Figure 4.3. Each DE consists of 100 independently trained MLPs, each with identical data. The prediction score for the RO is the average of the prediction scores output by the component MLPs. The AU-ARC shows a performance improvement. The results also show that the

DE generally performs well in terms of the ARC. The AU-ARC for the DE shows the best combined results in Table 4.2 and Table 4.4.

The DE is also found to positively affect signature classification with the RO. In our proposed model, any machine learning method can be used for the classification model as long as the RO is feasible. However, we conclude that the DE is the best choice for this experiment. In addition to the measured results, the advantage of using DE is that it extends MLP. MLP is a kind of DNN, and DNNs continue to make remarkable progress in terms of applications. Therefore, this signature classification model with an RO is also expected to benefit from the future development of MLPs and DNNs.

Table 4.4: AU-ARC (%) between the MLP and DE.

| Dataset | Features | MLP | DE |
|---|---|---|---|
| AAD | MCF | 99.98 $\pm$ 0.02 | 99.98 $\pm$ 0.02 |
| MAD | MCF | 99.42 $\pm$ 0.36 | **99.58 $\pm$ 0.32** |

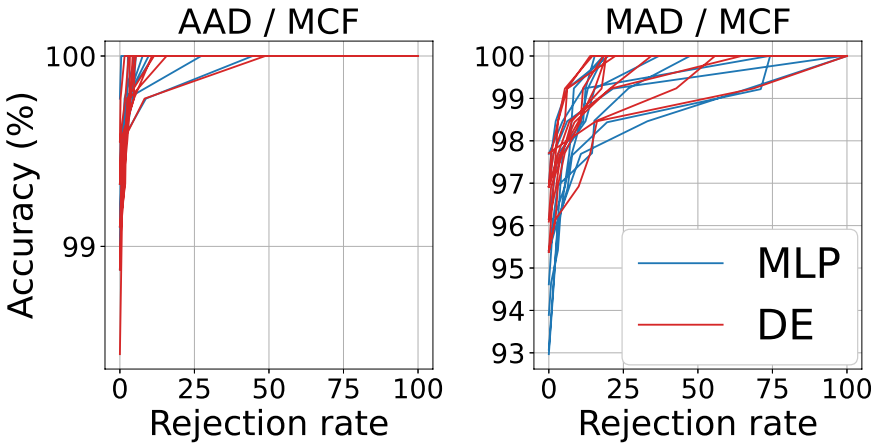


Figure 4.3: ARC confirms the improvement achieved by the RO with the DE.

### Analysis of the expert's point of view

Through interviews with experts, we design the WMFs by assuming that all the information in the *msg* and the information from the web is important. The values of the weight parameters of the trained models denote the feature importance in classifying signatures, so the features with large weights are considered by experts.

We analyze the feature importance in binary classification tasks (important vs nonimportant), where we merge the high and middle labels of the security importance level into one class (important). We apply the MCFs and linear SVM to the AAD and MAD for our analysis. A comparison of the classification model weights learned by linear SVM shows which factors among *5-tuple*, *metadata*, *classtype*, *msg*, and *reference* are considered more important. The average of the weights learned in each fold of stratified 10-fold cross-validation is calculated.

We can determine to which of the five elements of the signature each weight belongs. Figure 4.4 shows the cumulative frequency graphs of the classification model weights of the five elements. The horizontal axis shows the ranks of the absolute values of the weights. A comparison of the two figures shows that the important features are different depending on the type of dataset. The best and second-best features in the AAD are elements of *metadata*. Additionally, in third place is an element of *classtype*, and in fourth place is an element of the 5-tuple. The elements of *msg* appear in the 6th position and later, but the elements of *reference* do not appear in the top 20. On the other hand, the weights of the top eight features in the MAD are elements of *msg*. After the top nine, *reference* elements appear, and after the top 17, 5-tuple elements appear. *metadata* and *classtype* elements do not appear in the top 20 at all.

The features with high weights were consistent with the features identified as important in the expert interviews. We find that, unlike if-then rules, experts pay attention to *msg* and *reference* in manual classification. *msg* and the external information from *reference* are similar to natural-language information. If these are the dominant perspectives in manual classification, then it is likely that NLP methods can be applied.

### Analysis of valid features

We designed WMF based on the assumption that *msg* and *reference* are important based on the results of interviews with experts. To confirm the validity of this assumption, we analyze valid features through more detailed experiments. Under the conditions described above and the MCF, we performed the same experiment with 31 combinations of all the elements in the signature (*5-tuple*, *metadata*, *classtype*, *msg*, and *reference*) for MAD. The results are shown in Table 4.5. For notational convenience, *5-tuple* is abbreviated as 5t, *msg* as ms, *metadata* as mt, *reference* as rf, and *classtype* as cl. The value with the highest average BACC for each machine learning model is shown in bold. Underlines indicate the highest value in the machine learning model for each number of elements used.

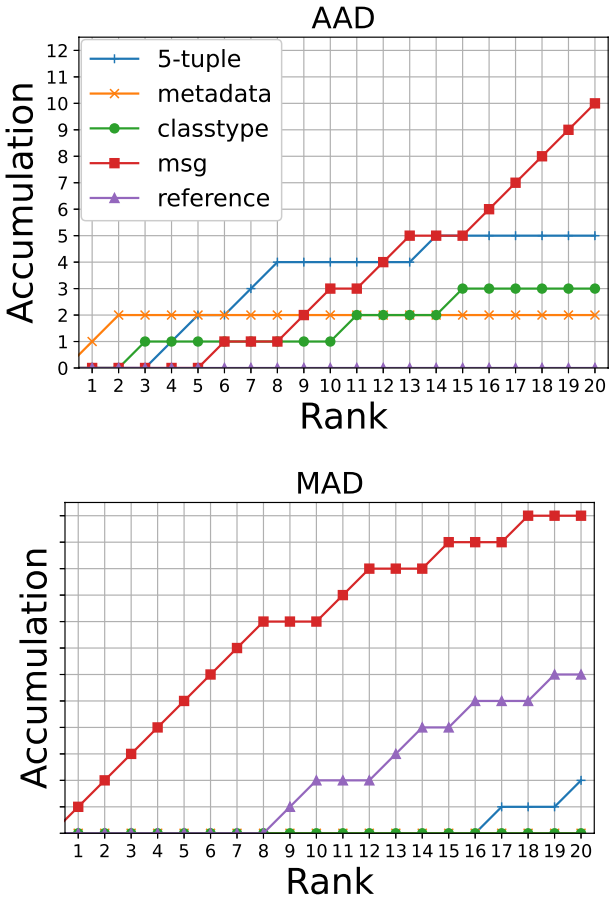


Figure 4.4: Feature analysis: the cumulative number of elements with high weights for each feature in the rankings on the horizontal axis.

Table 4.5: Detailed performance evaluation of the use of MCF in MAD

| Elements † | SVM | MLP | DT | RF | NB |
|---|---|---|---|---|---|
| 5t | 46.92 ± 7.87 | 44.66 ± 7.58 | 43.09 ± 5.97 | 45.22 ± 6.18 | 37.10 ± 6.64 |
| mt | 49.95 ± 5.27 | 50.00 ± 5.38 | 50.86 ± 6.01 | 51.44 ± 5.80 | 34.55 ± 4.08 |
| cl | 57.48 ± 7.87 | 56.92 ± 7.59 | 57.48 ± 7.87 | 56.64 ± 8.01 | 43.70 ± 5.23 |
| ms | 85.04 ± 4.83 | 87.38 ± 5.70 | 85.07 ± 8.31 | 83.32 ± 7.14 | 83.01 ± 7.68 |
| rf | 80.81 ± 7.48 | 82.66 ± 5.96 | 80.58 ± 7.81 | 80.72 ± 8.03 | 76.74 ± 4.48 |
| 5t,mt | 52.80 ± 5.18 | 55.16 ± 6.17 | 50.67 ± 7.77 | 49.91 ± 6.25 | 36.39 ± 6.53 |
| 5t,cl | 61.13 ± 8.44 | 59.67 ± 8.37 | 57.59 ± 10.6 | 58.10 ± 9.54 | 44.30 ± 9.28 |
| 5t,ms | 86.77 ± 4.71 | 85.22 ± 5.93 | 84.66 ± 6.89 | 76.25 ± 6.14 | 82.87 ± 7.87 |
| 5t,rf | 86.37 ± 7.04 | 86.23 ± 6.15 | 81.65 ± 6.36 | 82.14 ± 5.35 | 77.29 ± 4.93 |
| mt,cl | 55.73 ± 5.39 | 57.53 ± 8.04 | 56.59 ± 6.85 | 54.21 ± 7.86 | 42.43 ± 7.18 |
| mt,ms | 78.21 ± 6.12 | 80.73 ± 6.87 | 83.40 ± 6.63 | 80.75 ± 7.58 | 82.91 ± 7.25 |
| mt,rf | 83.74 ± 5.12 | 84.03 ± 7.65 | 84.64 ± 5.85 | 84.32 ± 5.44 | 75.47 ± 8.14 |
| cl,ms | 84.35 ± 4.86 | 86.23 ± 5.83 | 84.27 ± 6.08 | 83.25 ± 5.43 | 82.89 ± 8.24 |
| cl,rf | 86.16 ± 6.72 | 84.68 ± 7.19 | 85.76 ± 7.43 | 85.52 ± 7.91 | 77.16 ± 5.34 |
| ms,rf | 88.87 ± 5.97 | 89.19 ± 5.02 | 87.49 ± 7.04 | **88.38 ± 7.05** | **85.59 ± 6.80** |
| 5t,mt,cl | 56.61 ± 6.78 | 55.85 ± 8.45 | 54.70 ± 10.82 | 52.73 ± 7.85 | 43.61 ± 8.96 |
| 5t,mt,ms | 79.76 ± 6.76 | 81.40 ± 6.14 | 81.69 ± 8.33 | 76.54 ± 8.03 | 82.67 ± 7.45 |
| 5t,mt,rf | 83.47 ± 8.75 | 84.85 ± 7.57 | 83.46 ± 6.41 | 81.45 ± 4.89 | 73.63 ± 9.28 |
| 5t,cl,ms | 85.31 ± 5.04 | 84.13 ± 6.37 | 82.28 ± 6.88 | 76.99 ± 5.50 | 82.68 ± 8.35 |
| 5t,cl,rf | 86.56 ± 6.92 | 84.95 ± 6.37 | 80.64 ± 6.22 | 82.58 ± 6.17 | 77.81 ± 6.70 |
| 5t,ms,rf | **88.94 ± 6.18** | **89.47 ± 6.34** | 86.68 ± 7.45 | 83.49 ± 5.71 | 85.37 ± 6.84 |
| mt,cl,ms | 80.82 ± 5.83 | 81.07 ± 5.74 | 83.07 ± 6.64 | 81.68 ± 6.57 | 82.72 ± 7.87 |
| mt,cl,rf | 84.37 ± 4.82 | 85.89 ± 5.67 | 84.61 ± 7.80 | 82.80 ± 7.68 | 75.55 ± 7.35 |
| mt,ms,rf | 87.12 ± 6.79 | 86.79 ± 6.27 | 86.79 ± 7.68 | 84.41 ± 6.69 | 84.74 ± 6.93 |
| cl,ms,rf | 87.88 ± 5.47 | 89.28 ± 5.23 | 86.72 ± 7.09 | 87.34 ± 7.52 | 85.07 ± 7.82 |
| 5t,mt,cl,ms | 81.62 ± 6.13 | 81.04 ± 5.89 | 82.02 ± 7.32 | 71.80 ± 5.69 | 82.42 ± 8.01 |
| 5t,mt,cl,rf | 83.86 ± 7.95 | 85.65 ± 7.73 | 83.14 ± 8.21 | 81.19 ± 6.83 | 73.87 ± 9.01 |
| 5t,mt,ms,rf | 87.65 ± 6.27 | 86.76 ± 6.39 | 86.38 ± 9.14 | 84.03 ± 5.59 | 84.46 ± 7.07 |
| 5t,cl,ms,rf | 88.79 ± 5.31 | 87.58 ± 6.54 | 87.52 ± 6.92 | 85.81 ± 6.53 | 84.78 ± 7.80 |
| mt,cl,ms,rf | 86.53 ± 6.29 | 87.28 ± 6.63 | **89.40 ± 4.73** | 84.73 ± 6.17 | 84.52 ± 7.48 |
| 5t,mt,cl,ms,rf | 86.82 ± 6.35 | 86.45 ± 6.99 | 86.68 ± 6.96 | 82.38 ± 4.42 | 84.18 ± 7.57 |

†5tuple(5t), msg(ms), metadata(mt), reference(rf), classtype(cl)

We can confirm that *msg* and *reference* contribute significantly to performance improvement. The performance of all machine learning models is highest when *msg* and *reference* are included. Comparison by the number of elements converted to features shows that the performance of feature sets that include *msg* and *reference* tends to be higher. Comparing the five elements alone, Linear-SVM, Multilayer Perceptron, DT, and RF perform better than *msg*, *reference*, *classtype*, *metadata*, and *5-tuple*, in that order. For Naive Bayes, the top three are *msg*, *reference*, and *classtype*, in the same order as the other machine learning models.

From the above, we conclude that the assumption that *msg* and *reference* are important is valid. *msg* and *reference* are natural language elements, and NLP methods can be applied to them. NLP is one of the fields that is rapidly advancing with the rise of deep learning. For example, the BERT model [10] learned from large language corpora could be applied. BERT has been applied to many tasks with excellent results and may be applicable to [11, 12, 121] and signature classification.

## 4.4 Concluding Remarks

In this chapter, three features proposed as one element of the machine learning-based IDPS signature classification models were evaluated through experiments. SFs and KFs were designed based on if-then rules, and WMFs were designed based on the results of interviews with experts. WMF used the idea of expanding its information content by combining tf-idf and web scraping. A signature classification model was constructed by combining machine learning models and features, and evaluation experiments were conducted on AAD and MAD. By using the combined SF and KF features, the AAD was able to classify with high accuracy, but the MAD was only able to classify with relatively low accuracy. However, we confirmed that using features combining SF and WMF improved the performance of MAD. Through the analysis of effective features, we confirmed the validity of the assumptions and WMFs obtained from interviews with experts.

The analysis showed that *msg* is the most effective method for classifying signatures in MAD, followed by *reference*. Further performance improvement can be expected by using general-purpose language models and word embedding models used in natural language processing.

In this chapter, the machine learning model with RO was trained using the signatures obtained for one year and six months. Because time series were not considered, words not included in the training did not appear as unknown words at the time of testing. No problem that would significantly degrade the classification accuracy of the test data was observed. In the long term, it is expected that the information contained in the signatures will change with changes in software

information and types of malicious communications. If new words that the machine learning model has not yet learned appear in the test data, they may degrade the classification accuracy.

# Chapter 5

# A machine learning system that cooperates with an expert

## 5.1 Problem setup

We discuss considerations for automating our target task with machine learning. Three challenges are encountered when applying a classification model by machine learning to the real world: (a) security incidents caused by classification errors, (b) high annotation costs, and (c) classification accuracy decreases due to domain shifts. The details of each are as follows.

(a) **Security incidents caused by classification errors**: Classification errors can lead to improper IDPS configurations, which may cause security incidents. Security incidents can significantly damage the social credibility of an organization and sometimes cause fatal damage. As in the medical field, mechanisms that reduce classification errors, such as the reject option, are needed for our field.

(b) **High annotation costs**: Only experts with knowledge and experience can perform signature classification. In other words, collecting labeled signatures for training is not easy. It is necessary to train the classification model on a limited dataset.

(c) **Classification accuracy decreases due to domain shifts**: Signatures are periodically generated to keep up with new cyberattacks. This causes a domain shift, and there is concern that signature classification models will not be able to effectively classify new signatures. Figure 5.1 shows a simple analysis of this issue. We have developed a time-stamped signature dataset (Section3.4.2). In this dataset, we measured BACC using the following two holdout methods. (i) Ignoring timestamps (blue bars in Figure 5.1). (ii) Considering timestamps; new signatures were classified by a classification model trained on old signatures (red bars in Figure 5.1). If no domain shift has occurred, there should be little differences

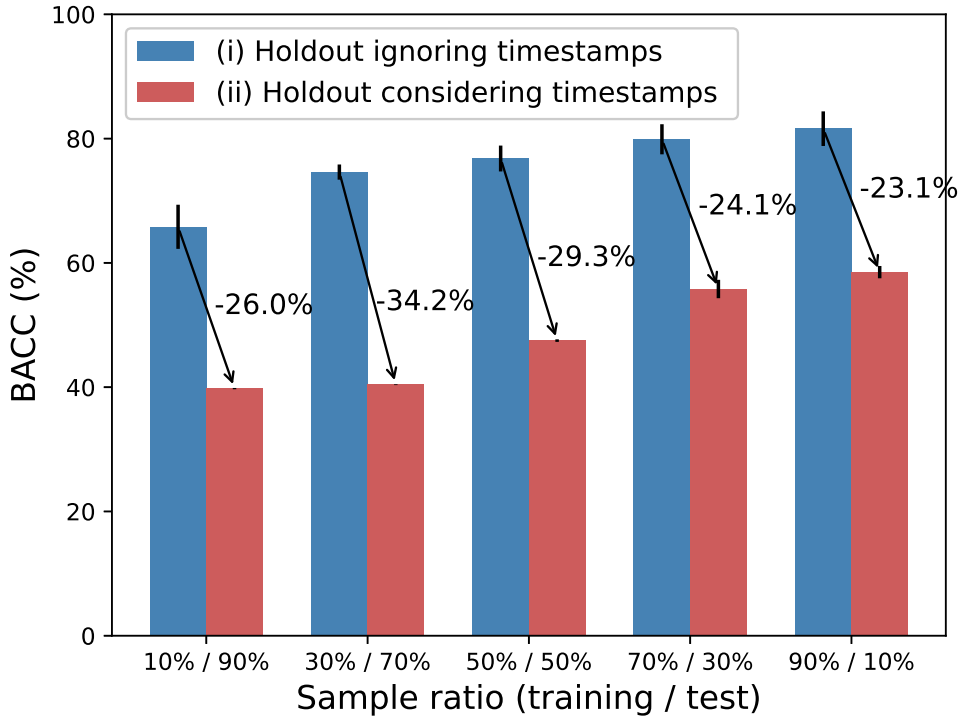


Figure 5.1: Experimental results showing the accuracy degradation induced in the IDPS signature classification model: the classification of new signatures with a model trained on old signatures resulted in accuracy degradation. A multilayer perceptron was used for the machine learning model, and the feature design and hyperparameters were the same as those in the experiment in Section 5.3.1.

between the BACCs of these cases. However, the BACC is lower for the time series split case. In other words, domain shift occurs; the simple classification models are ineffective in real situations. We need a mechanism to ensure that a classification model can keep up with new signatures.

The signature classification task is similar to the challenges of applying machine learning in the medical field; (a) security incidents caused by classification errors and (b) high annotation costs are common challenges. Our signature classification task is characterized by the challenge concerning (c) classification accuracy decreases due to domain shifts. In the medical field, classification targets are usually observation data from the human body, which do not change significantly even if the time series changes. However, signatures are generated as new cyberattacks are created, so they change significantly.

## 5.2 Signature classification with active learning

In this section, we propose an active learning system that uses the IDPS signature classification model with features proposed in Chapter 4 as one of the elements for overcoming the three challenges mentioned above: (a) security incidents caused by classification errors, (b) high annotation costs, and (c) classification accuracy decreases due to domain shifts. We choose uncertainty sampling to reduce misclassification as the acquisition function. Figure 5.2 shows an overview of a proposed system.

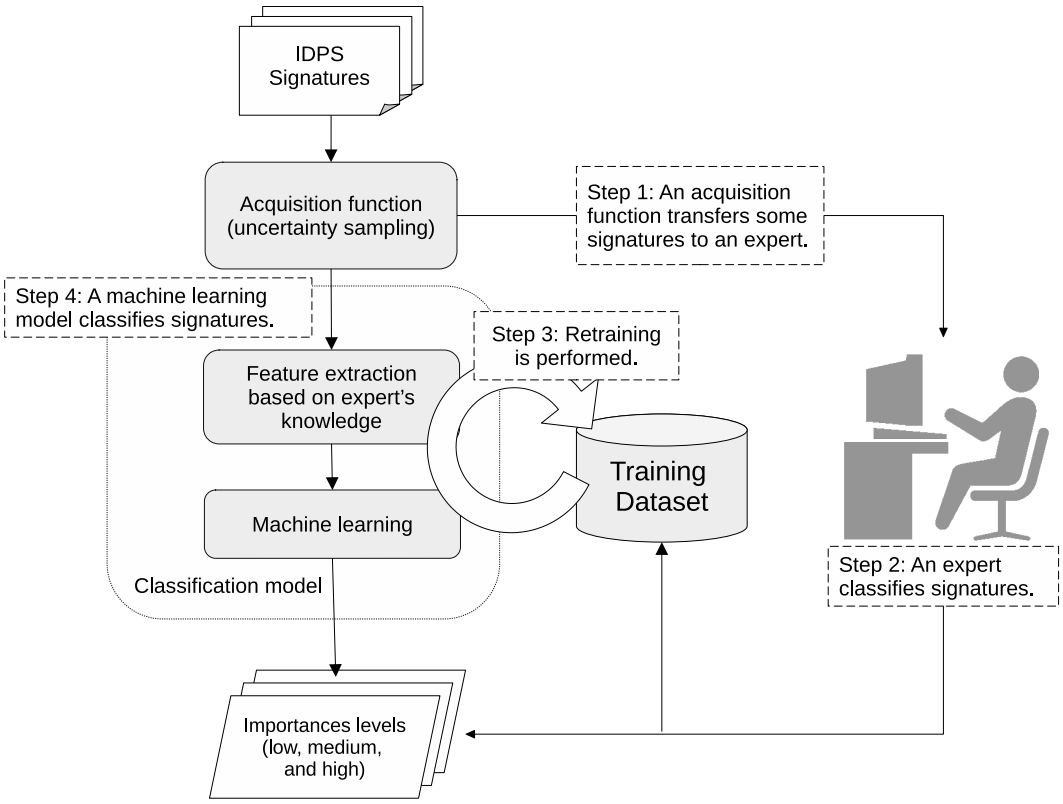


Figure 5.2: An overview of a proposed system based on active learning. First, the signatures whose importance levels are difficult to determine are transferred to an expert. The expert determines the importance of those signatures. The signatures are added to the training dataset along with their importance labels. After retraining, the machine learning model classifies the remaining signatures.

### 5.2.1 Procedures for applying active learning

We present notations showing the processing steps of the proposed active learning-based system. The key to understanding the procedure is how to build the training dataset and the set of signatures to be classified for each discrete time step $t \in \{0, 1, ..., \}$. Let $s^{(t)} \in \mathbb{N}$ be the number of annotations set by the expert at each time step $t$. $s^{(t)}$ is a hyperparameter that the system user sets while considering the annotation cost. At each time step $t$, the top $s^{(t)}$ signatures resulting from uncertainty sampling are annotated and added to the training dataset. After the classification model is retrained, the signatures that were not annotated are classified.

Let $\mathcal{X}^{(t)}$ be the space of the signatures at time $t$. The set of signatures sampled from $\mathcal{X}^{(t)}$ is $X^{(t)} = \{x_i^{(t)}\}_{i=1}^{N^{(t)}}$. $N^{(t)} \in \mathbb{N}$ is the number of signatures given at time $t$, e.g., the number of signatures distributed by the IDPS developer. The space of the importance labels assigned to the signatures is constant regardless of the time series, with $y \in \mathbb{Y} = \{1, ..., C\}$. The indices of the signatures manually annotated by experts at time $t$ are $\mathcal{I}_{train}^{(t)} \subset \{1, .., N^{(t)}\}$, and the indices of the signatures automatically classified by the machine learning model are $\mathcal{I}_{test}^{(t)} \subset \{1, ..., N^{(t)}\}$. Let $\mathcal{A}^{(t)} = \{x_i^{(t)}, y_i^{(t)} | i \in \mathcal{I}_{train}\}$ be the dataset labeled by the experts at time $t$. Let $\mathcal{D}_{train}^{(0)} = \{x_i^{(0)}, y_i^{(0)}\}_{i=1}^{N^{(0)}}$ be the initial training dataset. The training dataset at time $t$ is constructed by accumulating a labeled dataset $\mathcal{A}^{(t)}$ as follows.

$$\mathcal{D}_{train}^{(t)} = \mathcal{D}_{train}^{(0)} \cup \mathcal{A}^{(1)} \cup, ..., \cup \mathcal{A}^{(t)} = \mathcal{D}_{train}^{(t-1)} \cup \mathcal{A}^{(t)}$$

Let $w^{(t)}$ be the parameter of the classifier trained from $\mathcal{D}_{train}^{(t)}$. Let $P(y|x, w^{(t)})$ be the membership probability of sample $x$ as predicted by the classifier trained on $\mathcal{D}_{train}^{(t)}$. The output label of the classifier is calculated as

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x, w^{(t)}).$$

The procedure of the system at step $t$ is summarized in Algorithm 1. For effective active learning, it is important to design an acquisition function that selects annotation targets from unlabeled data. Our proposed system uses an uncertainty sampling strategy. The uncertainty estimation function $\upsilon : \mathcal{X}^{(t)} \rightarrow \mathbb{R}$ is used for uncertainty sampling. For the classification model to consistently achieve high accuracy, it is crucial to design the function $\upsilon$. The function $\upsilon$ should output higher values for signatures with more uncertain importance labels and lower values for more certain samples. The simplest example of $\upsilon$ is entropy. Entropy is calculated as follows:

$$\upsilon_H(x) = -\sum_{c \in \mathcal{Y}} P(y = c|x, w^{(t)}) \cdot \log(P(y = c|x, w^{(t)})).$$

**Algorithm 1** The procedure of the classification model at $t$

1: Given a dataset $X^{(t)} = \{x_i^{(t)}\}_{i=1}^{N^{(t)}}$ and $s^{(t)}$.
2: $\gamma^{(t)} \leftarrow \max_{[s^{(t)}]}(\{\upsilon(x_i^{(t)}) | x_i^{(t)} \in X^{(t)}\})$
3: $\mathcal{I}_{train}^{(t)} \leftarrow \{i | i \in \{1, ..., N^{(t)}\} \wedge \upsilon(x_i^{(t)}) \geq \gamma^{(t)}\}$
4: The experts annotate $X^{(t)}$ of $\mathcal{I}_{train}$ to develop a labeled dataset $\mathcal{A}^{(t)}$.
5: $\mathcal{D}_{train}^{(t)} \leftarrow \mathcal{D}_{train}^{(t-1)} \cup \mathcal{A}^{(t)}$
6: Train the classification model's parameter $w^{(t)}$ with $\mathcal{D}_{train}^{(T)}$.
7: $\mathcal{I}_{test}^{(t)} \leftarrow \{i | i \in \{1, ..., N^{(t)}\} \wedge \upsilon(x_i^{(t)}) < \gamma^{(t)}\}$
8: $X_{test}^{(t)} \leftarrow \{x_i^{(t)} | i \in \mathcal{I}_{test}^{(t)}\}$
9: Predict the classes in $X_{test}^{(t)}$ with the classification model trained in the above step.

Let $\max_{[n]}(\cdot)$ be a function that returns the $n$-th highest value among the given values. At the time $t$, the proposed system transfers the top $s^{(t)}$ uncertain signatures quantified by the function $\upsilon$ to the experts for annotation. The uncertainty threshold for selecting such signatures is as follows:

$$\gamma^{(t)} = \max_{[s^{(t)}]}(\{\upsilon(x_i^{(t)}) | x_i^{(t)} \in X^{(t)}\})$$

The indices $\mathcal{I}_{train}$ of the samples transferred to the experts at time $t$ and the index set $\mathcal{I}_{test}$ of the samples classified by the machine learning model are as follows:

$$\begin{aligned} \mathcal{I}_{train} &= \{i | i \in \{1, ..., N^{(t)}\} \wedge \upsilon(x_i^{(t)}) \geq \gamma^{(t)}\} \\ \mathcal{I}_{test} &= \{i | i \in \{1, ..., N^{(t)}\} \wedge \upsilon(x_i^{(t)}) < \gamma^{(t)}\} \end{aligned}$$

The experts annotate the signatures whose indices are contained in $\mathcal{I}_{train}^{(t)}$ to form $\mathcal{A}^{(t)}$. $\mathcal{A}^{(t)}$ and the training dataset $\mathcal{D}_{train}^{(t-1)}$ are combined into $\mathcal{D}_{train}^{(t)}$. Then, $\mathcal{D}_{train}^{(t)}$ is used to train $w^{(t)}$. After performing training, the target signature dataset $X_{test}^{(t)} = \{x_i^{(t)} | i \in \mathcal{I}_{test}\}$ is classified by the machine learning model. The above process is repeated with $t$ incremented each time a signature is distributed.

### 5.2.2 Uncertainty estimation performance improvement

There are two approaches to improving the performance of the proposed system. One is to investigate machine learning algorithms and feature designs with better classification and reject option performance. The second is to find a better method for preferentially transferring signatures with a high risk of misclassification to the expert using uncertainty sampling. In other words, to find a calibration method that brings the predicted probability vector output by the classification model

closer to the actual probability. We investigate the latter since we have already performed the former approach in Chapter 4.

We evaluate the performance of the proposed system when MC-Dropout[18] or a DE[19] is used for the classification model to improve the system's performance in Section 5.3.3. MC-Dropout is a method in which dropout[112], generally used during training, is also used during the inference process. The final output is the average of the probability vectors calculated from multiple feedforward steps. A DE is a classifier consisting of multiple neural networks. [114] conducted a study that examined the combination of active learning and uncertainty estimation. In the field of image recognition, this research experimentally showed that while MC-Dropout was adequate, the DE gave better results. However, in the experiments described below, we show that MC-Dropout performs better than the DE on the dataset used in this dissertation.

Both MC-Dropout and DE require multiple feedforward steps during inference. Assuming that the common variable $K$ represents the number of feedforward steps of each method, the prediction probabilities are as follows.

$$P(y|x, w^{(t)}) = \frac{1}{K} \sum_{k=1}^{K} P(y|x, w_k)$$

In the case of MC-Dropout, $w_k$ is the parameter of the neural network that is changed by the dropout operation at the $k$-th feedforward step. In the case of DE, $w_k$ is the parameter of the $k$-th neural network.

MC-Dropout and the DE can use Bayesian active learning by disagreement (BALD) [122] as the uncertainty estimation function in addition to entropy. Entropy and BALD are the two most popular acquisition functions in active learning-based uncertainty sampling. BALD determines the mutual information content between data points and weights $w_k$. This measure is the entropy of the probability vector output by the classification model minus the average conditional entropy for a given weight. The BALD function is as follows.

$$\upsilon_{BALD}(x) = \upsilon_H(x) - \lambda(x)$$

$\lambda(x)$ is the average conditional entropy for a given weight, which is given as follows.

$$\lambda(x) = \frac{1}{K} \sum_{k=1}^{K} \sum_{c=1}^{C} -P(y = c|x, w_k) \log P(y = c|x, w_k)$$

## 5.3 Evaluation of a proposed machine learning system

In this chapter, we evaluate the proposed system using the TMAD collected in cooperation with experts (Chapter 3). We also define our evaluation metric that matches the problem of classifying signatures in the real world with the imbalance between classes.

### 5.3.1 Experimental settings

#### An evaluation metric

In this section, we describe our evaluation metric. The typical top-1 classification accuracy and BACC measures, which do not consider the timing of data generation, are not appropriate for measuring the degree to which the problem in this dissertation is solved. It is impossible to measure the improvement of challenge (a)-(c) using Top-1 classification accuracy or BACC, typical metrics for machine learning. We define our metric, which takes the following three points into account.

- **Requirement for the challenge (a)**: It must be possible to quantify the performance of the RO. As in AU-ARC, the performance measure needs to be raised by forwarding samples that would otherwise be misclassified to an expert.
- **Requirement for the challenge (b)**: It must be possible to compare the classification accuracy on the labeled training datasets with the same number of samples. The number of training data is, in other words, the number of annotations by the experts. Furthermore, the number of annotations is the number that the system transfers to the experts. Its evaluation metric must be calculated on the condition that the number of transfers from the machine learning system to the expert is specified.
- **Requirement for the challenge (c)**: It must be a measure of overall performance throughout the time series. It must be computed assuming a labeled data set organized in time-series order.

In addition to these requirements, we also need to consider the imbalance of the class distributions that appear in the signature dataset. The class distribution imbalance is shown in Table 3.2.

The simulations evaluate the proposed system on a set of expert-labeled datasets $\mathcal{D} = \{\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, ..., \mathcal{D}^{(t)}, ..., \mathcal{D}^{(T)}\}$. $\mathcal{D}^{(t)} = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^{N^{(t)}}$ is a dataset consisting of the signatures generated at time $t$and their labels. Let $X_y^{(t)} = \{x_i^{(t)} | x_i^{(t)} \in$

$X^{(t)} \wedge y_i^{(t)} = y \wedge i = 1, ..., N^{(t)}\}$ be a subset of $X^{(t)}$ that consists only of signatures labeled $y$. The system prompts experts to label some signatures, retrains them, and classifies the rest of the signatures, repeating the sequence in discrete time order.

We define the co-balanced accuracy (CO-BACC) as an evaluation metric for this problem as follows:

$$\text{CO-BACC}(t; s^{(t)}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{\sum_{t'=1}^{t} \sum_{x \in X_y^{(t')}} \beta(x, y, t)}{\sum_{t'=1}^{t} |X_y^{(t')}|}$$

$$\beta(x, y, t) = \min(\mathbb{1}(y = \hat{y}) + \mathbb{1}(\upsilon(x) > \tau^{(t)}), 1) \tag{5.1}$$

The CO-BACC is a metric to be maximized, and it is calculated given $s^{(t)}$, the number of signatures transferred to the experts at each time step $t$. Through $t = 1, ..., T$, we assume that the signatures transferred to the experts are correct. This evaluation metric was inspired by the ARC used in the RO. The ARC allows us to visualize the tradeoff between the rejection rate and the classification accuracy when the rejected samples are considered correct. The classification accuracy of the ARC does not account for imbalance, but our CO-BACC is developed with this idea in mind.

Equation 5.1 shows that CO-BACC improves by either rejecting samples that misclassify or improving classification accuracy. Therefore, a comparison of the challenge (a) can be made. CO-BACC can compare methods on the same $s^{(t)}$, allowing the number of training data for the classification model to be fixed and allowing comparisons regarding the challenge (b). It is also possible to measure the degree of improvement of the challenge (c) since it is a comprehensive index over the time series.

One experiment is a simulation in which an expert and a classification model collaborate to classify signatures while $t = 1, ..., 23$. The CO-BACC is then calculated and plotted at each time step $t$. Let $r$ be the acquisition rate for each step $t = 1, ..., 23$. We perform a simulation for each $r = 10\%, 11\%, ..., 50\%$ and calculate the corresponding CO-BACC. If the number of samples to be acquired is not divisible, the decimal point is rounded down. For example, in the case of $r = 30\%$, $s^{(1)}$ is $\lfloor 283 \times 30\% \rfloor = 84$. The experiment is run 50 times with the same parameters, and the average value is used as the result.

### Feature sets

In this experiment, the features of the proposed system are those that combine only 5-tuple of SFs and WMFs (Figure 5.3). Section 4.3.2 results, see table 4.5, since the combination of these features shows the best classification performance.

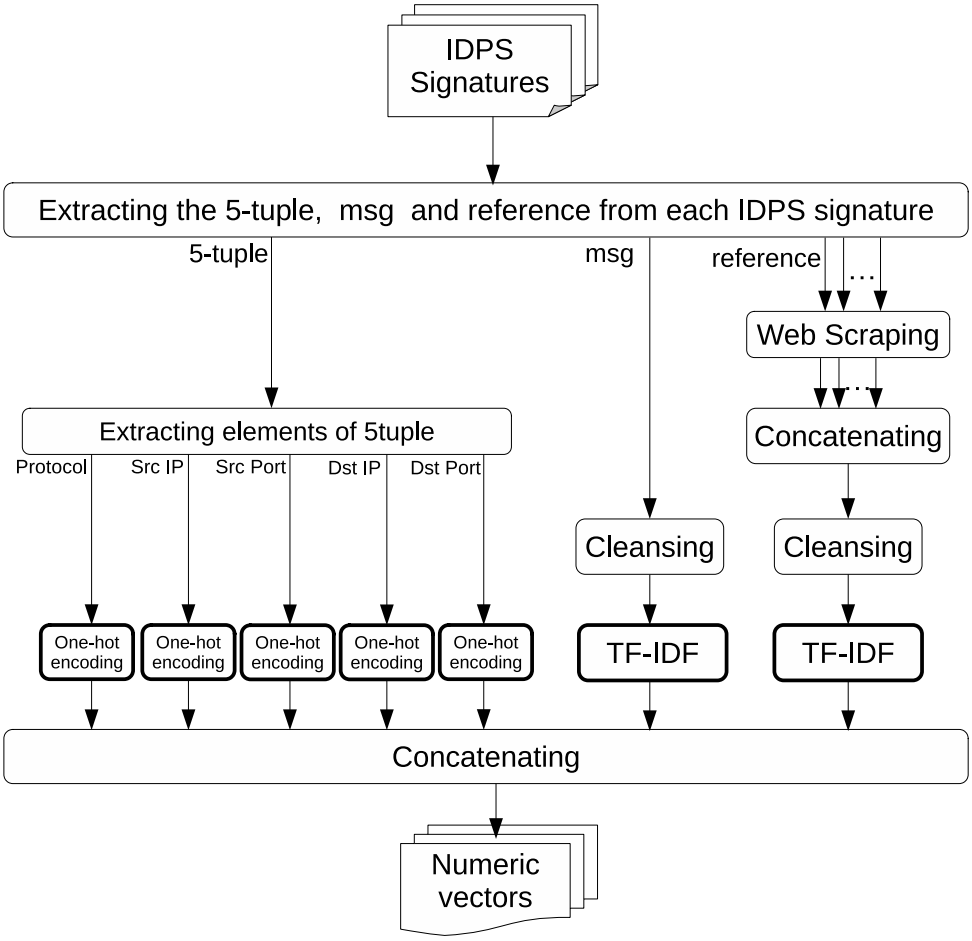


Figure 5.3: The feature extraction process of the experiments in Chapter 5

### Comparative methods

The classification model in this experiment is implemented based on a neural network as a machine learning model. The architecture of the neural network is a three-layer MLP with the following hyperparameters. A three-layer structure with an intermediate layer containing 100 nodes is trained using the error backpropagation method. The activation function for all nodes is a ReLU. L2 regularization is used to suppress overlearning. The regularization parameter is set to 0.0001. The optimization method is Adam with default parameters ($\alpha = 0.0001, \beta_1 = 0.9, \beta_2 = 0.99, and \epsilon = 10^{-8}$)[120]. The neural network training process is terminated when the loss value for the training data is less than 0.0001 from the minimum value at least 10 times (early stopping). The maximum number of epochs is set to 200.

In our experiment, the following six systems are implemented and compared

to confirm the effectiveness of the proposed system.

- **MLP-Random**: A neural network is used alone as the classification model, and random sampling is applied as the uncertainty estimation function.
- **MLP-Entropy**: A neural network is used alone as the classification model, and entropy is applied as the uncertainty estimation function.
- **DE-Entropy**: DE are employed as the classification model, and entropy is used as the uncertainty estimation function.
- **DE-BALD**: DE are employed as the classification model, and BALD is used as the uncertainty estimation function.
- **MCD-Entropy**: MC-Dropout is used when inferring neural networks, and entropy is used as the uncertainty estimation function.
- **MCD-BALD**: MC-Dropout is used when inferring neural networks, and BALD is used as the uncertainty estimation function.

The DE has 100 members. All hyperparameters are common among them, and the only randomness concerns the initial values of the weights and the choices of minibatches. The MC-Dropout probability is set to 0.5, and the feedforward step is performed 100 times.

### 5.3.2 Experimental results

In this section, we conduct experiments with the above datasets, evaluation metrics, and six systems and make comparisons. The first two simplest systems (MLP-Random and MLP-Entropy) are compared. Next, we validate the proposed improvements yielded by introducing the uncertainty estimation method to deep learning (DE-Entropy, DE-BALD, MCD-Entropy and MCD-BALD).

#### Comparing MLP-Random and MLP-Entropy

From left to right, Figure 5.4 shows the results obtained in the experiment with acquisition rates of 10%, 30% and 50%. The horizontal axis is $t$, and the vertical axis shows the CO-BACC at the corresponding step $t$. At all acquisition rates, MLP-Entropy outperforms MLP-Random in most of the steps $t$. In particular, MLP-Entropy outperforms MLP-Random even when the amount of training data is small and $t$ is small. MLP-Entropy can efficiently transfer samples to experts that are either appropriate for the training data or prone to errors.

Figure 5.5 shows the final CO-BACC values obtained for all acquisition rates tested. The acquisition rate leads to the number of samples submitted to the

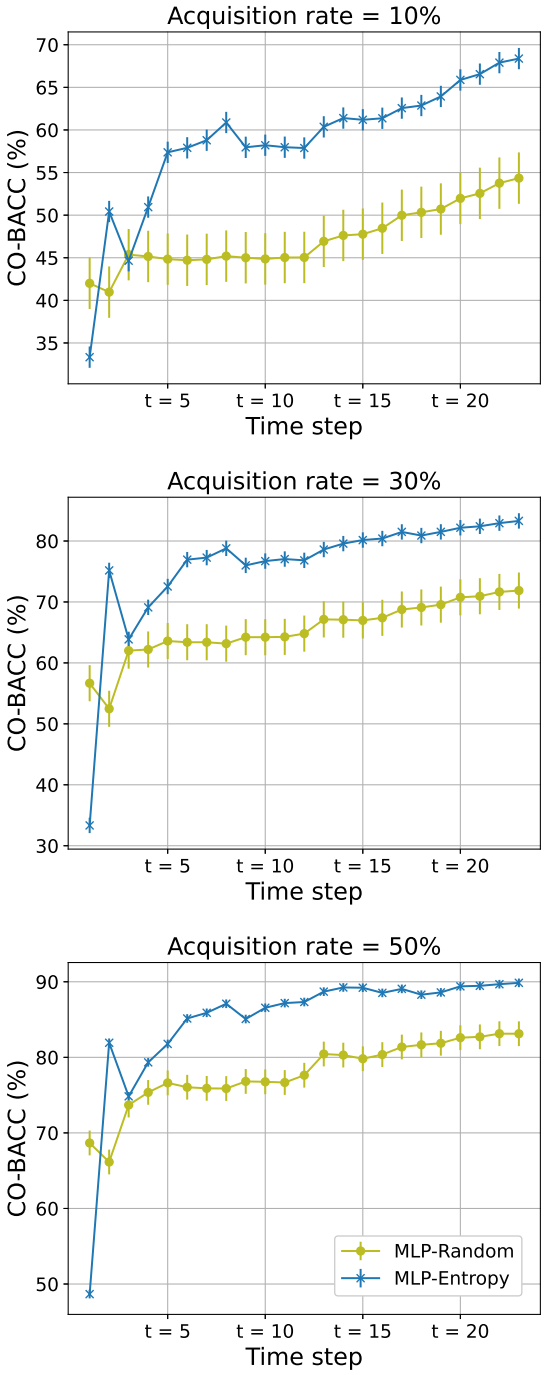


Figure 5.4: Experimental results comparing the performance of the proposed system with entropy-based uncertainty sampling (MLP-Entropy) with that of an MLP with a random acquisition function (MLP-Random).

expert at each step; i.e., it represents the size of the cost paid by the expert. It is an essential aspect of whether the proposed system works at any of the acquisition rates. Compared to MLP-Random, MLP-Entropy confirms its superiority in all cases, ranging from approximately 7% to 14% improvements. This indicates that the usability of the proposed system is high due to the wide range of acquisition rates the user can set.

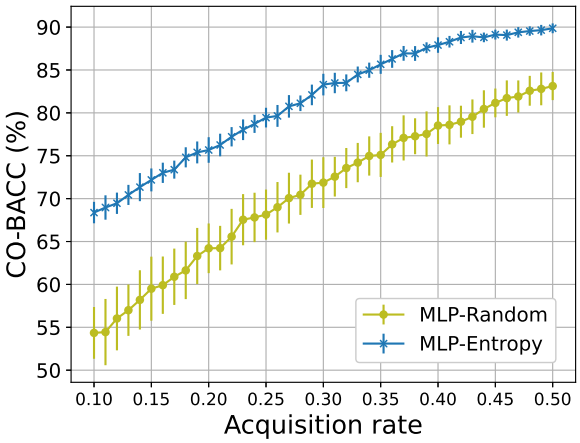


Figure 5.5: CO-BACCs at time step $t = 23$ (last step) for all acquisition rates.

In summary, MLP-Entropy works well in online signature classification tasks. Performing active learning with a simple neural network and the most primitive uncertainty sampling function as the acquisition function is effective for real-world situations.

### Verification of the proposed improvements

We incorporate MC-Dropout and the DE into the proposed system to verify the resulting performance improvements. Figure 5.6 shows the experimental results in the same style as Figure 5.4 but with a different system described. Figure 5.7 also shows the final CO-BACCs, as in Figure 5.5. It can be seen that the performance of the combined MCD systems, especially MCD-BALD, is generally higher than that of the combined DE systems and MLP-Entropy. MCD-BALD achieves a CO-BACC increase under relatively low numbers of transfers to experts with acquisition rates of 10% and 30% and at earlier stages in the time series. Eventually, the CO-BACCs of many systems converge to similar values, but MCD-BALD appears to be slightly dominant (Figure 5.7).

In [114], image recognition benchmarks were used, and DE performed better than MC-Dropout. In our experiments, however, MC-Dropout is preferred. The

image recognition and signature classification tasks differ significantly in the architecture of the neural network used for the classification model. The former uses a large neural network such as a convolutional neural network (CNN) to capture the complexity of images, while the latter uses a simple MLP. Inference with MC-Dropout can be interpreted as inference with different neural networks, i.e., pseudo DE. Although diversity among the neural network members is considered essential for DE [109, 110], MC-Dropout exhibits more diversity under the conditions of this experiment. This may be due to the positive effect of MC-Dropout on uncertainty sampling for relatively small neural networks. We suspect that the simple MLP, which classifies signatures, has fewer parameters than the CNN and thus cannot exhibit the diversity that results from random initial values. MC-Dropout performs better than the DE, which have been validated mainly on image recognition benchmarks and have been reported to perform well, so this is a surprising and valuable finding.

### 5.3.3 Analysis

The analysis focuses on MC-Dropout, which is the best-performing method. All values observed in the analysis are averages obtained over 50 trials.

#### Class distribution of the acquired samples

Since the experimental results show that MCD-BALD is superior, we analyze its behavior. First, we check the distribution of the importance levels of the acquired samples through simulations. We compare the MLP-Entropy values, MCD-BALD values and expected values (Random) for $r = 10\%, 30\%, 50\%$. The results are shown in Table 5.1. MCD-BALD acquires more samples with "medium" labels than the other approaches. The "medium" sample is the second minority class and is harder to predict as "medium" than the majority class ("low"). The acquisition of these samples and their forwarding to the experts may reduce the number of errors. The dataset has three importance labels: "low", "medium", and "high", which are ordered. The "medium" label is in the middle of these three labels and may be uncertain for the classification model since its importance is uncertain even for experts.

#### Time series in the distribution in the training dataset

Imbalance within the training dataset is an undesirable property that increases the concern that the classification model learns to neglect minority classes. The class distribution in the training dataset for each time step $t$ reveals that the proposed

system mitigates the imbalance issue. Figure 5.8 shows the training dataset distributions observed over time for $r = 10\%, 30\%, 50\%$. The vertical axis indicates the percentage of signatures with "medium" or "high" importance in the dataset, and the horizontal axis indicates the time step. The green lines indicate the expected values of the class distribution when the signatures are transferred randomly. MLP-Entropy (blue lines) and MCD-BALD (red lines) also have roughly higher percentages of minority samples than the random acquisition strategy. This trend is particularly pronounced in the latter half of the period, indicating that the imbalance issue is rapidly dissipating.

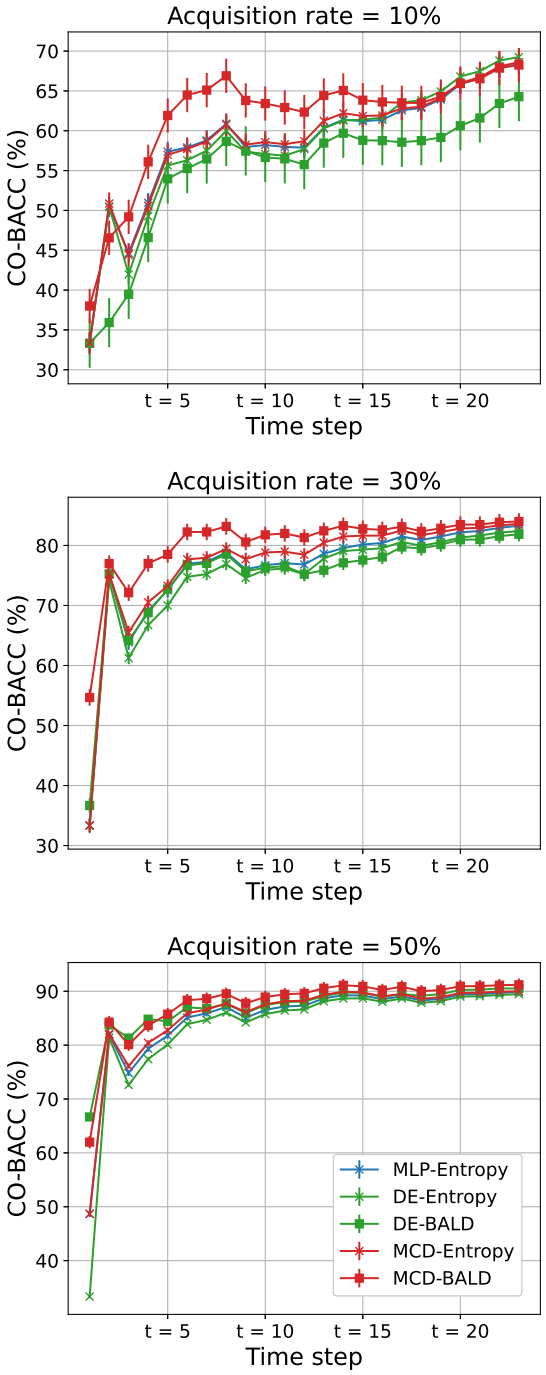


Figure 5.6: Comparison among the performances of the plain proposed system (MLP-Entropy) and four proposed systems combined with the uncertainty estimation method (MCD-Entropy, MCD-BALD, DE-Entropy, and DE-BALD)

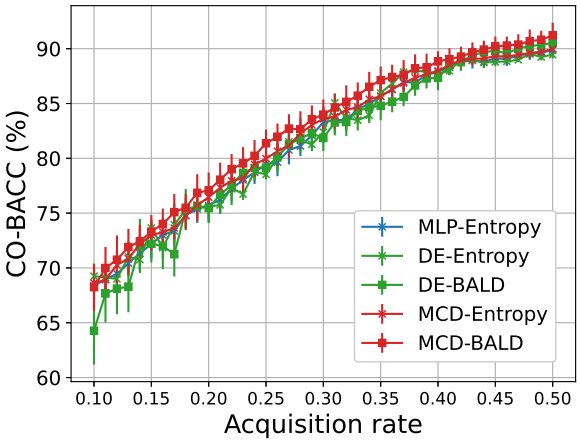


Figure 5.7: CO-BACCs obtained at step $t = 23$ for the plain proposed system (MLP-Entropy) and the four proposed systems (MCD-Entropy, MCD-BALD, DE-Entropy, and DE-BALD) combined with the uncertainty estimation method.

Table 5.1: Class distributions of the acquired samples

| Acquisition rate | 10% | | | 30% | | | 50% | | |
|---|---|---|---|---|---|---|---|---|---|
| Method \Class | low | medium | high | low | medium | high | low | medium | high |
| MLP-Entropy | 471.3 | 223.5 | 32.2 | 1683.0 | 469.0 | 59.0 | 3071.4 | 549.5 | 72.0 |
| MCD-BALD | 470.9 | 221.2 | 34.9 | 1660.4 | 494.5 | 56.0 | 3043.9 | 576.8 | 72.3 |
| Random | 668.1 | 62.5 | 9.7 | 2004.3 | 187.5 | 29.1 | 3340.5 | 312.5 | 48.5 |

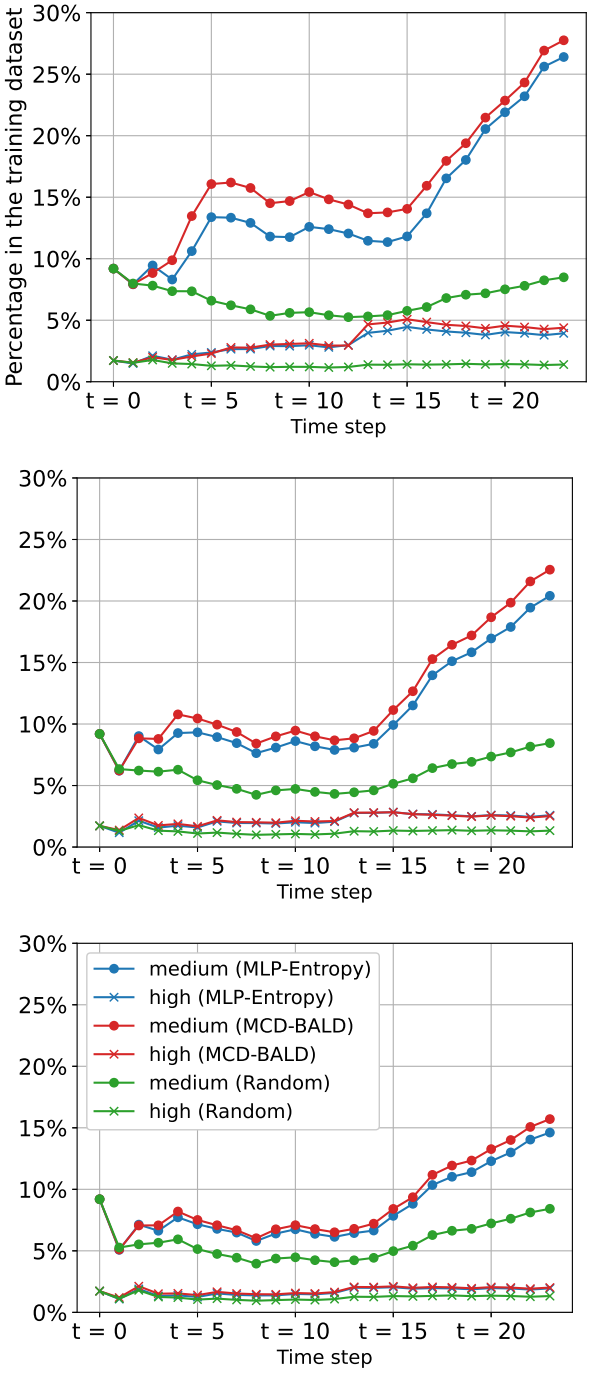


Figure 5.8: The class distributions in the training dataset for each time step. These are the percentages of data samples (signatures) with "medium" or "high" importance labels in the training dataset. The training data imbalance issue is suppressed in the cases of MLP-Entropy (blue lines) and MCD-BALD (red lines) compared to the case where the signatures are acquired randomly (green lines).

### 5.3.4 Discussion

The key idea of our proposed system is to overcome the multiple challenges that arise when applying signature classification models within an active learning framework to real-world situations. We explain why the proposed system can overcome the challenges (a)-(c).

- **Discussion about (a) Security incidents caused by classification errors:** Transferring signatures to experts in the active learning acquisition function corresponds to a reject option, a function that avoids classification errors by canceling uncertain classifications. CO-BACC represents the integrated classification accuracy when a human and a machine learning model share the classification task. Specifically, only when the machine learning model makes an error mistake is it considered incorrect. In the Figure 5.4, the machine learning model can reduce the number of classification errors since the value is superior to that of the random transfer.

  In particular, MCD-BALD transferred more "medium" signatures to the experts (Table 5.1). Experts prefer that the "medium" signature, considered the boundary between "low" and "high" signatures, be transferred preferentially. Analysis using Table 5.1 suggests reducing misclassification by manually checking samples on the boundaries.

- **Discussion about (b) High annotation costs:** In the active learning process, which is the basis of the system, the system picks up data that are considered helpful for training and trains on a smaller set of labeled data by asking humans to label the data. Whether the annotation cost has been reduced can be determined by comparing the classification accuracy between methods trained with the same annotation cost. The acquisition rate in the Figure 5.4 corresponds to the annotation cost. Even with the same annotation cost, the proposed system (MLP-Entropy) shows better CO-BACC, indicating that it mitigates the annotation cost issue.

  Analysis using Table 5.8 showed that active learning reduced the imbalance in the training data. In other words, more minority class samples can be added to the training data. This leads to fewer classification errors for minority classes, suggesting that training can be done more efficiently with less annotation cost.

- **Discussion about (c) Classification accuracy decreases due to domain shifts:** The classification model is also naturally updated sequentially to keep up with the periodically generated signatures. The proposed system's mechanism of adding new data to the training data as needed is thought to

allow it to keep up with new data. In other words, it can reduce the influence of domain shift. Figure 5.4 of the experimental results shows that the proposed system's CO-BACC of both MLP-Entropy and MLP-Random increases gradually. After time step $t = 3$, MLP-Entropy has a more rapid CO-BACC increase than MLP-Random until the halfway point. Assuming that the better performance is better at following new data, MLP-Entropy, which is based on active learning, is better at following new data.

The proposed system is based on active learning, which allows the system and humans to cooperate. The system reduces the burden on experts by performing some of the manual signature classifications. In addition, while the system is in operation, humans classify signatures as usual, which leads to an expansion of the effective training dataset and contributes to improving the classification accuracy by the system. Thus, the system works for the benefit of people, and their actions also benefit the system.

Feedback from the system to the human can also be provided, but that is a topic for future work. An example of feedback is when a human and a system perform classification in parallel, and the system communicates disagreements to the human. However, to provide feedback to humans, it is necessary to guarantee the reliability of the feedback. Therefore, we must first assume that the expert's determination is true and verify the estimation performance of the system.

## 5.4 Concluding remarks

To classify signatures generated over time, we proposed a system in which a human and a machine learning model cooperate by applying active learning. We defined the CO-BACC as a performance measure that considers the class imbalance issue, the time series of the given signatures, and the behavior of the experts asked to make classification decisions. Experiments conducted on the TMAD showed that the attained CO-BACC was higher with active learning-based uncertainty sampling than with random signature sampling. Furthermore, it was confirmed that incorporating MC-Dropout, a deep learning-based uncertainty estimation method, into the proposed system further yielded improved performance. The analysis showed that incorporating MC-Dropout resulted in the transfer of more signatures with "medium" importance to the experts and a gradual alleviation of the imbalanced nature of the training data.

In this chapter, we proposed a system based on active learning in cooperation with experts. The system can be applied to other data by replacing the feature extraction process. There are other tasks in network security operations where data are generated periodically and classified by experts. For example, software vulnerability information, such as CVE, is released periodically, and experts may

decide whether to classify this information as necessary. The proposed system can be applied to such tasks and can be widely applied to other tasks as well.

# Chapter 6

# Conclusion

This dissertation addressed the challenges in building a machine learning system that cooperates with humans to classify intrusion detection and prevention system (IDPS) signatures. Conclusions and future works are described below.

First, we collected three datasets with experts from real network operating organizations. The first dataset was named *automatically annotated dataset (AAD)*. AAD consists of classified signatures using if-then rules, which are scripts for automatically classifying signatures. Experts prefer to classify signatures using if-then rules whenever possible. Hence, we know that experts code the if-then rule scripts based on all their explicit knowledge. The second dataset was named *manually annotated dataset (MAD)*. The experts constructed the MAD by manually annotating signatures that if-then rules could not classify. All signatures that can be classified based on explicit knowledge are included in the AAD. Therefore, the signatures in the MAD are classified based on the experts' tacit knowledge. The third dataset was named *time-series manually annotated dataset (TMAD)*. Signatures in TMAD were manually classified and given the date and time they were distributed. AAD, MAD, and TMAD consist of $4,465$, $1,300$, and $7,577$ labeled signatures, respectively. Until now, the IDPS signature classification task dataset has not been made publicly available for reasons of conventions and security. We also investigated their IDPS signature classification procedures through interviews with experts and described them. With these works, we have completed the necessary preliminaries for our research.

Second, IDPS signature classification models were built and analyzed using AAD and MAD. In this dissertation, two approaches were used for feature design that is easy to analyze. The first approach is based on the if-then rule. In this approach, two types of features, *symbolic features (SFs)* and *keyword features (KFs)* were designed. The second approach was designed through interviews with experts. The interviews revealed that natural language elements in the signatures and information on the Internet are important. We designed a set of features

named *web information and message features (WMFs)*. In our experiments, we combined these features with several machine learning methods to build a classification model. The experiments revealed the following points.

- In the AAD, the maximum value of BACC was measured to be 95.69 percent for the feature set that combined SFs and KFs. This indicates that the design of these features aimed at reproducing the if-then rule was successful.
- In MAD, the SFs and KFs achieved a maximum BACC of only 59.59 percent, while the combination of SFs and WMFs achieved a maximum performance of 86.82 percent. This indicates that the WMFs better capture the human decision criteria.
- WMFs performed well when the area under the accuracy-rejection curve (AU-ARC) was measured as a reject option (RO) performance.
- The performance of the RO was improved by using deep ensembles (DE), an uncertainty estimation method in deep learning.
- The analysis showed that when experts manually classify signatures, they focus first on *msg* and second on *reference* (information scraped from the Internet).

The above work makes it possible to construct an IDPS signature classification model, a human-collaborative machine earning system module. Among the proposed features, WMF is superior in classification accuracy and RO performance, and we should adopt WMF as a feature. Incorporating DE into the classification model improved reject option performance, indicating that it is also an effective method for building a cooperative system with experts. The analysis shows that external information referenced by *msg* and *reference* is important, confirming the validity of the WMF design, which was inspired by the results of interviews with experts.

Third, to overcome three challenges in applying signature classification models to real-world applications (a) security incidents caused by classification errors, (b) high annotation cost, and (c) classification accuracy decrease due to domain shifts, we proposed and evaluated an active learning-based system with uncertainty sampling as the acquisition function. Whenever the proposed system receives signatures sent like a subscription service, it forwards some of them to an expert for annotation. The annotated signatures are added to the training data, retrained, and the remaining signatures are classified. The signatures transferred to the experts are determined based on uncertainty sampling. The proposed system aims to improve the accuracy of uncertainty sampling by using the uncertainty

estimation method in deep learning. The following results were obtained through simulation experiments based on the actual operation using the dataset.

- The proposed system outperforms random sampling. This indicates that the active learning method worked as intended.
- The performance was improved using Monte Carlo dropout (MC-Dropout), a deep learning uncertainty estimation method. Although we confirmed the performance improvement of RO with DE in Chapter 4.3 of the research, MC-Dropout showed more promising results.
- MC-Dropout was found to transfer more "medium" signatures to the experts, which are considered highly important to people but also tricky for them to make decisions.
- Experiments showed that the proposed system reduces the imbalance in the training data.

The above work shows that an active learning-based system can efficiently classify signatures in cooperation with experts. Incorporating MC-Dropout was experimentally demonstrated to be effective in increasing the ability of experts to transfer signatures with a high risk of incorrect answers. Furthermore, it mitigates imbalances in the training data, and more signatures with "medium" signatures on the importance boundary are forwarded to experts, suggesting that it positively impacts the classification model's training.

Future work in machine learning IDPS signature classification is the need to train classification models for each organization. A classification model trained for one organization may perform much worse for another organization because the importance determination of signatures is based on the IDPS used by the organization to which the expert belongs. These facts indicate that each organization needs to perform its annotation, which is even more expensive. Other than active learning, there are other ways to deal with this problem, such as weakly supervised learning [123] that can learn with a small number of labeled data sets, such as transfer learning [124] and semi-supervised learning [125]. For transfer learning, models trained in different organizations can be fine-tuned for other organizations to improve performance. This is in line with the characteristics of a network operation organization, where sensitive information cannot be exchanged.

There is also a need to address tasks other than IDPS signature classification. The three challenges addressed in Chapter 5, (a) security incidents caused by classification errors, (b) high annotation costs, and (c) classification accuracy decreases due to domain shifts, also appear in other tasks in network operations. The processing procedures of the system proposed in this dissertation can be applied to

different types of data. However, careful investigation, including feature design, is required for each individual task. Each organization has a different aspect of network operation, and the event itself, the communication network, is complex. This makes it difficult to take an approach where many researchers explore one standard benchmark, for example, in image recognition tasks. This is because network tasks can be unique, and completing one does not necessarily mean that everything has been solved. The signature classification task addressed in this dissertation may yield different results if labeled signatures from different organizations are used. However, even in such a case, we must gain as much knowledge as possible by individually verifying each signature one by one. Due to the sensitivity of the information handled, it is not easy to exchange specific data between network operating organizations. Federative learning [126], in which individually trained models are merged after the fact, maybe a good approach. The ultimate task is to tackle other specific tasks individually and, finally, to find a general rule.

In this dissertation, we proposed a system based on active learning that allows the machine learning system to cooperate with experts. Specifically, the machine learning system performs a part of signature classification on behalf of the expert, and the expert assists in the continuous training of the system through his/her daily work. The collaboration in this dissertation refers to the efficient sharing of tasks between systems and humans. It is just one example of collaboration, and other collaboration cases are possible. In this dissertation, a human determination is treated as absolute, but in reality, humans can make errors in determination. In such cases, an interaction framework is needed in which systems and humans mutually exchange opinions to find a solution. We can consider the interaction between a system and a person as a collaboration. Another example of collaboration is the use of trained machine learning systems as supervisors to train novices. The search for better training methods for experts is a significant challenge for network operation, and cooperation in the form of education using machine learning systems has social significance. In addition to task sharing, problem solving through system-human interaction and training of experts using learned models are future challenges for machine learning technology to support network operations. We hope that the ideas and evaluation results in this dissertation will help solve signature classifica- tion problems as well as other tasks.

# Bibliography


[1] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):1199–12000, 2009.

[2] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, 2016.

[3] S. Das and M. J. Nene. A survey on types of machine learning techniques in intrusion prevention systems. In *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2296–2299, Mar. 2017.

[4] Licheng Jiao and Jin Zhao. A survey on the new generation of deep learning in image processing. *IEEE Access*, 7:172231–172263, 2019.

[5] Junhyung Kang, Shahroz Tariq, Han Oh, and Simon S. Woo. A survey of deep learning-based object detection methods and datasets for overhead imagery. *IEEE Access*, 10:20118–20134, 2022.

[6] Hung-yi Lee, Shang-Wen Li, and Thang Vu. Meta learning for natural language processing: A survey. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 666–684, Seattle, United States, Jul. 2022.

[7] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, Mar. 2021.

[8] Raouf Boutaba, Mohammad Ali Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada Solano, and Oscar Mauricio Caicedo Rendon. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9:1–99, 2018.

[9] Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials*, 21(2):1988–2014, 2019.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, Jun. 2019.

[11] Y. Iwasaki, A. Yamashita, Y. Konno, and K. Matsubayashi. Japanese abstractive text summarization using bert. In *International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 1–5, Nov. 2019.

[12] Z. Gao, A. Feng, X. Song, and X. Wu. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299, 2019.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901. Curran Associates, Inc., Dec. 2020.

[14] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021.

[15] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *International Conference on Machine Learning (ICML)*, Aug. 2017.

[16] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[17] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Pe-

ter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *arXiv:2107.03342*, 2021.

[18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, Jun. 2016.

[19] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413. Dec. 2017.

[20] H. Shahriar and W. Bond. Towards an attack signature generation framework for intrusion detection systems. In *IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 597–603, Nov. 2017.

[21] N. Fallahi, A. Sami, and M. Tajbakhsh. Automated flow-based rule generation for network intrusion detection systems. In *Iranian Conference on Electrical Engineering (ICEE)*, pages 1948–1953, May. 2016.

[22] S. Lee, S. Kim, S. Lee, J. Choi, H. Yoon, D. Lee, and J. Lee. Largen: Automatic signature generation for malwares using latent dirichlet allocation. *IEEE Transactions on Dependable and Secure Computing*, 15(5):771–783, Sep. 2018.

[23] C. Constantinides, S. Shiaeles, B. Ghita, and N. Kolokotronis. A novel online incremental learning intrusion prevention system. In *IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–6, Jun. 2019.

[24] P. R. Chandre, P. N. Mahalle, and G. R. Shinde. Machine learning based novel approach for intrusion detection and prevention system: A tool based verification. In *IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, pages 135–140, Nov. 2018.

[25] S. Tengl, Z. Zhang, L. Teng, W. Zhang, H. Zhu, X. Fang, and L. Fei. A collaborative intrusion detection model using a novel optimal weight strategy based on genetic algorithm for ensemble classifier. In *IEEE International*

*Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 761–766, May. 2018.

[26] P. Lotfallahtabrizi and Y. Morgan. A novel host intrusion detection system using neural network. In *IEEE Annual Computing and Communication Workshop and Conference (CCWC)*, pages 124–130, Jan. 2018.

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

[28] Monowar Bhuyan, Dhruba K Bhattacharyya, and Jugal Kalita. Towards generating real-life datasets for network intrusion detection. *International Journal of Network Security*, 17:675–693, Nov. 2015.

[29] Radhika Chapaneri and Seema Shah. A comprehensive survey of machine learning-based network intrusion detection. In Suresh Chandra Satapathy, Vikrant Bhateja, and Swagatam Das, editors, *Smart Intelligent Computing and Applications*, pages 345–356, Singapore, 2019.

[30] Tadeusz Pietraszek. Using adaptive alert classification to reduce false positives in intrusion detection. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *Recent Advances in Intrusion Detection*, pages 102–124, Berlin, Heidelberg, 2004.

[31] K. Alsubhi, E. Al-Shaer, and R. Boutaba. Alert prioritization in intrusion detection systems. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 33–40, Apr. 2008.

[32] F. M. Cortés and N. Gaviria Gómez. A hybrid alarm management strategy in signature-based intrusion detection systems. In *IEEE Colombian Conference on Communications and Computing (COLCOM)*, pages 1–6, Jun. 2019.

[33] Tadeusz Pietraszek. Using adaptive alert classification to reduce false positives in intrusion detection. In *Recent Advances in Intrusion Detection*, pages 102–124. Springer Berlin Heidelberg, 2004.

[34] Neminath Hubballi and Vinoth Suryanarayanan. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, 49:1–17, 2014.

[35] Natalia Stakhanova and Ali A. Ghorbani. Managing intrusion detection rule sets. In *European Workshop on System Security*, EUROSEC '10, pages 29–35, New York, NY, USA, Apr. 2010.

[36] F. Massicotte and Y. Labiche. An analysis of signature overlaps in intrusion detection systems. In *IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, pages 109–120, Jun. 2011.

[37] Piyawat Noiprasong and Assadarat Khurat. An ids rule redundancy verification. In *International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 110–115, Nov. 2020.

[38] Yoshihide Nakagawa, Yuta Kazato, and Yuichi Nakatani. Inspecting intrusion prevention system signatures for false blocking using set theory. In *IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, Jun. 2020.

[39] C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, Dec. 1957.

[40] C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

[41] Harish G. Ramaswamy, Ambuj Tewari, and Shivani Agarwal. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1):530 – 554, 2018.

[42] Viktor Losing, Barbara Hammer, and Heiko Wersing. Knn classifier with self adjusting memory for heterogeneous concept drift. In *IEEE International Conference on Data Mining (ICDM)*, pages 291–300, Dec. 2016.

[43] Heitor Murilo Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfahringer, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106:1–27, Oct. 2017.

[44] Jan Goepfert, Barbara Hammer, and Heiko Wersing. Mitigating concept drift via rejection. In *International Conference on Artitificial Neural Networks (ICANN)*. Springer, Oct. 2018.

[45] M. H. Waseem, M. S. A. Nadeem, A. Abbas, A. Shaheen, W. Aziz, A. Anjum, U. Manzoor, M. A. Balubaid, and S. Shim. On the feature selection methods and reject option classifiers for robust cancer prediction. *IEEE Access*, 7:141072–141082, 2019.

[46] Dongyun Lin, Lei Sun, Kar Ann Toh, Jing Bo Zhang, and Zhiping Lin. Biomedical image classification based on a cascade of an svm with a reject

option and subspace analysis. *Computers in Biology and Medicine*, 96:128–140, May. 2018.

[47] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *International Conference on Machine Learning (ICML)*, pages 5281–5290, Jun. 2019.

[48] Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *International Workshop on Machine Learning in Systems Biology*, volume 8 of *Proceedings of Machine Learning Research*, pages 65–81, 2009.

[49] M. R. Abbas, M. S. A. Nadeem, A. Shaheen, A. A. Alshdadi, R. Alharbey, S. Shim, and W. Aziz. Accuracy rejection normalized-cost curves (arnccs): A novel 3-dimensional framework for robust classification. *IEEE Access*, 7:160125–160143, 2019.

[50] Filipe Condessa, José Bioucas-Dias, and Jelena Kovačević. Performance measures for classification systems with rejection. *Pattern Recognition*, 63(C):437 – 450, 2017.

[51] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Comput. Surv.*, 54(9), Oct. 2021.

[52] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *International Conference on Machine Learning (ICML)*, pages 417–424, Jun. 2006.

[53] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. pages 399–407, Sep. 2017.

[54] Ju Nam, Sunggyun Park, Eui Jin Hwang, Jong Lee, Kwang-Nam Jin, Kun Lim, Thienkai Vu, Jae Sohn, Sangheum Hwang, Jin Mo Goo, and Chang Min Park. Development and validation of deep learning–based automatic detection algorithm for malignant pulmonary nodules on chest radiographs. *Radiology*, 290:180237, Sep. 2018.

[55] Jiacheng Wang, Zhaocai Chen, Liansheng Wang, and Qichao Zhou. An active learning with two-step query for medical image segmentation. In *International Conference on Medical Imaging Physics and Engineering (ICMIPE)*, pages 1–5, 2019.

[56] Vishwesh Nath, Dong Yang, Bennett A. Landman, Daguang Xu, and Holger R. Roth. Diminishing uncertainty within the training pool: Active learning for medical image segmentation. *IEEE Transactions on Medical Imaging*, 40(10):2534–2547, Nov. 2021.

[57] Rosa Figueroa, Qing Zeng-Treitler, Long Ngo, Sergey Goryachev, and Eduardo Wiechmann. Active learning for clinical text classification: Is it better than random sampling? *Journal of the American Medical Informatics Association (JAMIA)*, 19:809–16, Jun. 2012.

[58] Kevin De Angeli, Shang Gao, Mohammed Alawad, Hong-Jun Yoon, Noah Schaefferkoetter, Xiao-Cheng Wu, Eric B Durbin, Jennifer Doherty, Antoinette Stroup, Linda Coyle, Lynne Penberthy, and Georgia Tourassi. Deep active learning for classifying cancer pathology reports. *BMC bioinformatics*, 22(1):113, Mar. 2021.

[59] Michael Bloodgood and Chris Callison-Burch. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. pages 854–864, Uppsala, Sweden, Jul. 2010.

[60] Pei Zhang, Xueying Xu, and Deyi Xiong. Active learning for neural machine translation. In *International Conference on Asian Language Processing (IALP)*, pages 153–158, Nov. 2018.

[61] Álvaro Peris and Francisco Casacuberta. Active learning for interactive neural machine translation of data streams. In *Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium, Oct. 2018.

[62] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M. Mitchell. Competence-based curriculum learning for neural machine translation. 2019.

[63] Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. Active learning approaches to enhancing neural machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1796–1806, Online, Nov. 2020.

[64] Justin Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148, Jan. 2018.

[65] Yukang Gong, Dongyu Xue, Guohui Chuai, Jing Yu, and Qi Liu. Deepreac+: deep active learning for quantitative modeling of organic chemical reactions. *Chem. Sci.*, 12:14459–14472, 2021.

[66] Tom A. Young, Tristan Johnston-Wood, Volker L. Deringer, and Fernanda Duarte. A transferable active-learning strategy for reactive molecular force fields. *Chem. Sci.*, 12:10944–10955, 2021.

[67] Xiaoyu Zhang. Interactive patent classification based on multi-classifier fusion and active learning. *Neurocomputing*, 127:200–205, Mar. 2014.

[68] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330, Aug. 2017.

[69] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[70] Meelis Kull, Telmo M. Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electron. J. Statist.*, 11(2):5052–5080, 2017.

[71] Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12295–12305, Dec. 2019.

[72] Jize Zhang, Bhavya Kailkhura, and T Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*, Jul. 2020.

[73] Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. Local temperature scaling for probability calibration. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6889–6899, Oct. 2021.

[74] Takahiro Mimori, Keiko Sasada, Hirotaka Matsui, and Issei Sato. Diagnostic uncertainty calibration: Towards reliable machine predictions in medical domain. In Arindam Banerjee and Kenji Fukumizu, editors, *International*

*Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130 of *Proceedings of Machine Learning Research*, pages 3664–3672. PMLR, Apr. 2021.

[75] Xingchen Ma and Matthew B. Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In *International Conference on Machine Learning (ICML)*, pages 7235–7245, Jul. 2021.

[76] Kanil Patel, William H. Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. In *International Conference on Learning Representations (ICLR)*, May. 2021.

[77] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Jun. 2016.

[78] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations (ICLR) Workshops*, Apr. 2017.

[79] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4696–4705, Dec. 2019.

[80] Seonguk Seo, Paul Hongsuck Seo, and Bohyung Han. Learning for single-shot confidence calibration in deep neural networks through stochastic inferences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9030–9038, Jun. 2019.

[81] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning (ICML)*, pages 7034–7044, Jul. 2020.

[82] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 18237–18248, Dec. 2020.

[83] Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. Calibrated language model fine-tuning for in- and out-of-distribution data. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online, Nov. 2020.

[84] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. In *International Conference on Machine Learning (ICML)*, pages 26135–26160, Jul. 2022.

[85] Shohei Enomoto and Takeharu Eda. Learning to cascade: Confidence calibration for improving the accuracy and computational cost of cascade inference systems. In *Conference on Artificial Intelligence (AAAI)*, Feb. 2021.

[86] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Conference on Artificial Intelligence (AAAI)*, pages 2901–2907, Jan. 2015.

[87] Juozas Vaicenavicius, David Widmann, Carl R. Andersson, Fredrik Lindsten, Jacob Roll, and Thomas B. Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 3459–3467. PMLR, Apr. 2019.

[88] Michael Kranzlein, Nelson F. Liu, and Nathan Schneider. Making heads and tails of models with marginal calibration for sparse tagsets. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4919–4928, Nov. 2021.

[89] Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2019.

[90] Joaquin Quiñonero-Candela, Carl Rasmussen, Fabian Sinz, Olivier Bousquet, and Bernhard Schölkopf. *Evaluating Predictive Uncertainty Challenge*, volume 3944, pages 1–27. Apr. 2006.

[91] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[92] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research).

[93] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[94] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Jun. 2009.

[95] Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online, Nov. 2020.

[96] Yibo Hu and Latifur Khan. Uncertainty-aware reliable text classificationu. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*.

[97] Soham Dan and Dan Roth. On the effects of transformer size on in- and out-of-domain calibration. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2096–2101, Punta Cana, Dominican Republic, Nov. 2021.

[98] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Annual Conference on Computational Learning Theory (COLT)*, pages 5–13, Jul. 1993.

[99] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

[100] David Barber and Christopher M Bishop. Ensemble learning in bayesian neural networks. *Nato ASI Series F Computer and Systems Sciences*, 168:215–238, 1998.

[101] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688, Jun. 2011.

[102] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2348–2356, Dec. 2011.

[103] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning (ICML)*, pages 1683–1691, Jun. 2014.

[104] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, pages 1613–1622, Jul. 2015.

[105] Aurick Zhou and Sergey Levine. Training on test data with bayesian adaptation for covariate shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, Dec. 2021.

[106] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13991–14002, Dec. 2019.

[107] Fredrik K. Gustafsson, Martin Danelljan, and Thomas B. Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[108] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations (ICLR)*, Apr. 2020.

[109] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. In *Advances in Neural Information Processing Systems (NeurIPS)*, Dec. 2020.

[110] Thomas Elsken Chris C. Holmes Frank Hutter Yee Teh Sheheryar Zaidi, Arber Zela. Neural ensemble search for uncertainty estimation and dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*. Dec. 2021.

[111] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations (ICLR)*, Apr. 2022.

[112] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[113] Tomoyuki Myojin, Shintaro Hashimoto, and Naoki Ishihama. Detecting uncertain bnn outputs on fpga using monte carlo dropout sampling. In *International Conference on Artificial Neural Networks (ICANN)*, page 27–38, Berlin, Heidelberg, Sep. 2020.

[114] William H. Beluch, Tim Genewein, Andreas Nurnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In

*IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, Jun. 2018.

[115] Y. Yang. Research and realization of internet public opinion analysis based on improved tf - idf algorithm. In *International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES)*, pages 80–83, Oct. 2017.

[116] P. Sun, L. Wang, and Q. Xia. The keyword extraction of chinese medical web page based on wf-tf-idf algorithm. In *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 193–198, Oct. 2017.

[117] A. I. Kadhim. Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In *International Conference on Advanced Science and Engineering (ICOASE)*, pages 124–128, Apr. 2019.

[118] Joel Nothman, Hanmin Qin, and Roman Yurchak. Stop word lists in free open-source software packages. In *Workshop for NLP Open Source Software (NLP-OSS)*, pages 7–12, Melbourne, Australia, Jul. 2018.

[119] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. 16(1):321–357, Jun. 2002.

[120] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, May. 2015.

[121] H. Sohn and H. Lee. Mc-bert4hate: Hate speech detection using multi-channel bert for different languages and translations. In *International Conference on Data Mining Workshops (ICDMW)*, pages 551–559, Nov 2019.

[122] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

[123] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, Aug. 2017.

[124] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the Institute of Radio Engineers*, 109(1):43–76, Jan. 2021.

[125] Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109:373 – 440, 2019.

[126] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

# Publication List

川口英俊，中谷裕一，岡田将吾: 専門家の知見に基づいた特徴量設計による IDPS シグネチャ重要度分類, 情報処理学会論文誌 62 巻 9 号, 2021, pp-1575-1585,

Hidetoshi Kawaguchi, Yuichi Nakatani, and Shogo Okada: IDPS Signature Classification with a Reject Option and the Incorporation of Expert Knowledge, 21st IEEE International Conference on Machine Learning and Applications (ICMLA), 2022

Hidetoshi Kawaguchi, Yuichi Nakatani, and Shogo Okada: IDPS signature classification based on active learning with partial supervision from network security experts, IEEE Access, 2022

# Acknowledgements

First and foremost, I would like to thank Associate Professor Shogo Okada for accepting my proposed research project before entering the Japan Advanced Institute of Science and Technology (JAIST) and allowing me to conduct research. His precise guidance helped me grow as a researcher. Even though my paper was rejected several times due to my inadequacies, he encouraged me and patiently guided me through the research process. Thanks to Prof. Okada, I continued my research for three and a half years and finally finished my doctoral thesis.

Also, I would like to thank Mr. Yuichi Nakatani, the second author of all my published papers and my supervisor at my company. He had helped me in many aspects of my daily research activities, even before I started at JAIST. He is like a big brother to me, the closest and most supportive person I have ever had, as I also do research and development in the company.

I would also like to thank the experts in network security operations who assisted me in my research. I regret that I cannot disclose their names. Without them, I could not have started this research in the first place.

Next, I thank Associate Professor Kiyoaki Shirai for his guidance on my minor research project. I also thank Prof. Kunihiko Hiraishi, Prof. Shinobu Hasegawa, and Prof. Kokolo Ikeda for serving as members of my dissertation review committee. Their constructive comments helped me to improve my dissertation further. I would also like to thank Prof. Akio Kawabata of the Toyohashi University of Technology, a doctoral thesis reviewer from outside JAIST. He commented on my dissertation and encouraged me to consider enrolling in a doctoral program.

Finally, I would like to thank my father for his encouragement before I enrolled and my wife for her support in life. Thank you very much.