IDPS Signature Classification with a Reject Option and the Incorporation of Expert Knowledge

Hidetoshi Kawaguchi 1, 2, Yuichi Nakatani 1 and Shogo Okada 2

1 Network Innovation Center, Nippon Telegraph and Telephone (NTT), Tokyo, Japan
2 School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan

Abstract—As the importance of intrusion detection and prevention systems (IDPSs) increases, great costs are being incurred to manually manage signatures, which are malicious communication pattern files. Network security experts need to classify signatures by importance for an IDPS to work optimally. In this study, we propose and evaluate a machine learning signature classification model with a reject option (RO) to reduce the cost of setting up an IDPS. Experts classify some signatures with predefined if-then rules. 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 extended by web scraping from the messages and external attack identification system name. The third set is a language feature based on the keywords in the message text and the attack identification system name. We evaluate the effectiveness of the proposed model using two datasets, which consist of real signatures labeled by experts. One is a dataset that can be classified with an if-then rule. The other dataset consists of signatures that do not match the if-then rule. The experiments show that the proposed features are valid for the task of classifying the two datasets by using multiple machine learning classification models.

Index Terms—machine learning, feature engineering, reject option, real data, Snort, IDPS

1. INTRODUCTION

Intrusion detection and prevention systems (IDPSs) are security systems for computer networks. These systems perform several actions, for example, recording communication logs, alerting to the need for investigation, and preventing attacks on a network, when they detect malicious communications. An IDPS detects intrusions based on signatures, which are malicious communication pattern files [1]. We can set the actions of the IDPS for each signature.

Experts in network security management need to determine whether each action should be executed for signatures based on the importance labels assigned to each signature. Great costs are incurred to manage signature data manually.

To reduce the cost of classifying signatures, this study aims to build a signature classification model with a reject option (RO) using machine learning [2], [3]. The RO is a function used to cancel classification when the model’s output appears uncertain. Signature classification can cause catastrophic damage to communication networks if it fails. We should consider the cost of misclassification to be as high as in the medical field. It is a natural idea to add an RO to prevent classification errors as well as possible. To design features for classification, we refer to methods by which experts classify important. In general, experts classify signatures somewhat automatically. First, the experts classify signatures using an if-then rule that they construct. 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.

In this study, a single classification model is used to classify all signatures. Therefore, following the expert classification methods described above, we propose three sets of features. The first is a feature set that is subject to the conditions of an if-then rule. The second set is a feature set obtained based on the keywords in the message text and the attack identification system name. The third set is a language feature extended by web scraping from the messages and external attack identification systems described in the signatures.

We evaluate the effectiveness of the proposed model using two datasets, which consist of real signatures labeled by experts. One is a dataset that can be classified with an if-then rule. The other dataset consists of signatures that do not match the if-then rule. The experiments show that the proposed features are valid for the task of classifying the two datasets by using multiple machine learning classification models.

The contributions of this paper are summarized as follows: (i) We develop a real dataset for signature classification with experts in network security operational practice. (ii) We design an effective feature set to classify the signature patterns in classification tasks with an RO. (iii) We evaluate both the classification accuracy and the RO of the trained models. (iv) The analysis identifies and experimentally demonstrates practical machine learning algorithms for ROs and signature elements of interest to experts.

A. Related works

1) Reducing the management costs of IDPSs: Our study provides an approach for reducing the management costs of IDPSs. There are multiple approaches to such studies, including identifying duplicate signatures [4]–[6], detecting signatures that cause false alerts [7], and reducing the false alerts obtained from IDPSs [8]–[10]. 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.

To the best of our knowledge, the idea of performing automatic classification by using machine learning to classify signatures directly has not been explored. In this paper, the primary challenge in classifying signatures in machine learning is addressed. We design effective features for classification and evaluate and analyze them.

2) Reject option (RO): The RO was pioneered by Chow [2], [3]. The idea of dropping a classification according to certain criteria is widely used. An RO is helpful in fields where the impact of misclassification is significant, such as the medical field [11], [12].

In IDPS signature classification, the subject of this study, the cost of failure is as high as that in the medical field. 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. To mitigate such risks, using an RO in the classification model of IDPS signatures is a natural solution.

II. PROBLEM SETUP

A. Dataset

The evaluations in this paper are carried out on signatures labeled by experts engaged in network security management. The experts design if-then rules 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. First, the experts automatically classify signatures with the if-then rules. 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 automatically annotated dataset (AAD), and the latter is called the manually annotated dataset (MAD).

Table I shows the number of AAD and MAD samples prepared by the experts. Each signature is assigned one of three importance labels: low, medium, or high. Based on the importance level, the experts set an action of the IDPS for communications that match the signature.

B. Notation of signatures

The signatures in the AAD and MAD are written in the notation of the IDPS engine called Snort [13]. Fig. 1 shows a concrete example [14]. The first word “alert” is the action taken by the IDPS when the signature is matched. Because the experts set up actions based on importance, actions cannot be entered into the classification model. Features are extracted from the strings after the action.

“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. 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 Fig. 1 is an example of this.

*metadata* is an element that represents information in the key-value format. The “ruleset community” in Fig. 1 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 Fig. 1, it is described as “cve,1999-0197”. In this example, *reference* refers to Common Vulnerabilities and Exposures (CVE) 1999-0197.

*classtype* is a general group of malicious communications indicated by signatures. In Fig. 1, it is “attempted-recon”. The groups that *classtype* indicates are different from the importance levels judged by the experts.

C. If-then rule

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 *message* 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.

| AAD | MAD |
|-----|-----|
| 3,996 | 119 |
| 93 | 122 |
| 436 | 59 |
| 4,465 | 3,300 |

**Table I**

**Dataset of Signatures.**

In Fig. 1, Specific examples of IDPS signatures.
The if-then rule assigns importance labels only to signatures that match certain conditions. Signatures that do not qualify for keyword matching are assigned the label “unknown”.

III. A CLASSIFICATION MODEL WITH A REJECT OPTION

To construct a classification model with an RO, we propose symbolic features (SFs), keyword features (KFs), and web information and message features (WMFs). We also describe the behavior of the RO incorporated into the classification model. Fig. 2 shows the procedure of the proposed classification model. The SFs and KFs are designed with reference to if-then rules. The WMFs are designed with reference to interviews with experts.

A. Symbolic features

SFs are extracted from the 5-tuple, metadata, and classtype, each of which is extracted as a feature with one-hot encoding.

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.

B. Keyword features

KFs are designed for keyword matching on msg and reference in the if-then rule. 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.

C. Web information and message features

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. From the interviews, we learned the following:

- 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 expand on this information, we apply term frequency-inverse document frequency (TF-IDF), which is frequently used in natural language analysis [15], [16]. Web scraping is also used to perform feature extraction on the reference.

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 IDs in web systems such as the National Vulnerability Database (NVD) [17] and RedHat’s CVE Database [18]. Examples of signature-related information include the software targeted by a malicious communication, which is 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 [19] 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. TF-IDF is converted into numeric vectors composed of numerical values obtained by multiplying the tf and idf documents. tf represents the number of occurrences of a word in the document. idf expresses the
rarity of the word, i.e., the reverse document frequency. The \textit{idf} used when converting the test data to feature vectors with TF-IDF is the \textit{idf} calculated from 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.

D. If-then rule features and manual classification features

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.

E. Classification with a reject option

The proposed model is equipped with an RO. The RO means that the classification can be rejected depending on the value of the class’s prediction score.

The RO can be used on any classification model as long as a prediction score can be calculated. It is formulated as follows: Let \( x \in X \) be the input class and \( y \in Y = \{1, \ldots, K\} \) be the output class. Let \( S_y : X \rightarrow \mathbb{R} \) be a function that computes the prediction score of class \( y \) for a classifier. The classification model with an RO determines the final classification prediction, class \( \hat{y} \), as follows:

\[
\hat{y} = \begin{cases} 
\arg \max_{y \in Y} S_y(x) & \text{if } \max_{y \in Y} S_y(x) \geq \tau \\
\text{reject} & \text{otherwise}.
\end{cases}
\]

\( \tau \) is the threshold, which is the hyperparameter of the RO.

IV. EXPERIMENT

A. Experimental settings

1) Outline of the experiment: In this section, we confirm the classification accuracy and RO performance of the proposed classification model and analyze the validity of the feature design. Specifically, experiments are conducted on the following process: (1) We validate the proposed features (ITRFs and WMFs) on several traditional machine-learning models and compare the classification accuracy. (2) We evaluate the quantified RO performance with accuracy-rejection curves (ARCs) and the area under the ARC (AU-ARC) [20] and evaluate the classification accuracy. (3) We explore the performance improvement due to the RO by using a robust machine-learning model for the RO (deep ensemble [21]), which is said to better represent uncertainty. (4) We analyze the importance of the proposed features to identify the signature elements that the experts regard as important for evaluating the signature. The results of these experiments are described in the form given in Sections IV-B1, IV-B2, IV-B3, IV-B4.

2) Evaluation method: Due to the imbalanced dataset, we measure the balanced accuracy as the classification accuracy. In addition, to verify the performance of the RO, we plot an ARC [20]. 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. If the AU-ARC is included, it is possible to compare methods regardless of the threshold value \( \tau \). The experiments are performed with trained 10-fold cross-validation.

3) Machine learning: In experiments involving two different trained classification models, a linear SVM and a multi-layer perceptron (MLP), 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 [22]. The number of neighbors is 5.

The classification model of the 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 ramp function is selected as the activation function for all nodes. Overfitting is suppressed by L2 regularization. The regularization parameter is set to 0.0001. We use Adam [23] for the optimization of objective functions. The Adam parameters are set to the default values presented in [23]. Training is terminated if the loss value obtained for the training data is not less than 0.0001 for a minimum of 10 iterations.

4) Prediction score for the RO: 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.

B. Experimental results

1) Evaluation classification accuracy: Table II shows the balanced accuracy results of this experiment. The “dataset” column indicates the target dataset. The “features” column shows the features into which the signature of the dataset has been transformed. The columns with the names of the machine learning methods show the corresponding means and standard deviations of the balanced accuracy.

Both machine learning models achieved balanced accuracies of 95% or higher in the AAD experiments. Therefore, we assume that the ITRFs adequately express the if-then rules. On the other hand, the balanced accuracy of the ITRFs on the
MAD is lower than that on the AAD. Even though the ITRFs can sufficiently follow the if-then rules, they had difficulty in classifying the MAD.

To compare the ITRFs and MCFs, we confirm the experimental results on the MAD. The results show that the MCFs have higher balanced accuracy rates for both machine learning models. The performance of the linear SVM and MLP improve by 28.96% and 23.87%, respectively.

We evaluate the performance of the MCFs on the AAD. The results indicate that no machine learning models exhibit significant performance improvements. The WMFs in the MCFs seem to effectively capture the characteristics of manual classification.

2) Evaluation of RO performance: Table III shows the AU-ARC for the RO. The overall trend is similar to that of balanced accuracy. 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.

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.

3) Evaluation of a deep ensemble for the RO: In this section, we use a deep ensemble (DE) [21] 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 [2]. The DE is considered a method that better represents the uncertainty of deep neural networks (DNNs). Experiments have also shown the superior calibration capability of the DE, which is the ability to estimate the true probability [21], [24].

The analysis results are shown in table IV and Fig. 4. 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 II and table IV.

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.

4) 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 important. 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. Fig. 5 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

| Dataset | Features | Linear 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 (±0.81) | 95.18 (±0.36) |
|         | MCF      | 99.45 (±0.36) | 99.42 (±0.36) |

Fig. 3. Each ARC shows the result of 1 fold in stratified 10-fold cross-validation.

Fig. 4. ARC confirms the improvement achieved by the RO with the deep ensemble.
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 natural-language processing (NLP) methods can be applied.

V. CONCLUSION

We proposed a machine learning importance classification model with an RO for IDPS signatures. We designed SFs, KFs, and WMFs as features for this model. The SFs and KFs were designed based on if-then rules. The WMFs were designed based on interviews with experts. We proposed features by combining TF-IDF and web scraping to obtain WMFs. We developed real datasets, the AAD and MAD, consisting of signatures labeled with importance levels by real experts. We conducted experiments using two different machine learning models. The use of combined SFs and KFs classified the AAD with high accuracy. When we extracted and classified the features from the signatures of the MAD, we obtained only low accuracy. However, the use of combined SFs and WMFs yielded a significant performance improvement on the MAD. These trends in the classification accuracy were similar to the performance of the RO. The results also showed that DE might be promising for improving the performance of the RO. We also analyzed the validity of the WMFs by checking the weights of the classification model trained by linear SVM.

The experimental results showed that the message text and web-based feature expansion contribute to improving the classification accuracy and RO performance on the MAD. One future research direction will concern the use of advanced NLP techniques. It is also necessary to explore how to operationalize the classification model to handle daily signature generation.

REFERENCES

[1] S. Das and M. J. Nene, “A sudrevy on types of machine learning techniques in intrusion prevention systems,” in International Conference on Wireless Communications, Signal Processing and Networking (WISPNET), 2017, pp. 2296–2299.
[2] C. K. Chow, “An optimum character recognition system using decision functions,” IEEE Transactions on Information Theory, vol. EC-6, no. 4, pp. 247–254, 1957.
[3] C. Chow, “On optimum recognition error and reject tradeoff,” IEEE Transactions on Information Theory, vol. 16, no. 1, pp. 41–46, 1970.
[4] N. Stakhanova and A. A. Ghorbani, “Managing intrusion detection rule sets,” in European Workshop on System Security, 2010, pp. 29–35.
[5] F. Massicotte and Y. Labiche, “An analysis of signature overlaps in intrusion detection systems,” in IEEE/IP Conference on Dependable Systems and Networks (DSN), 2011, pp. 119–129.
[6] P. Noiprasong and A. Khurat, “An ids rule redundancy verification,” in International Joint Conference on Computer Science and Software Engineering (IJCSSE), 2019, pp. 1–6.
[7] Y. Nakagawa, Y. Kazato, and Y. Nakatani, “Inspecting intrusion prevention system signatures for false blocking setting theory,” in IEEE International Conference on Communications Workshops (ICC Workshops), 2020, pp. 1–6.
[8] F. M. Cortés and N. Gaviria Gómez, “A hybrid alarm management strategy in signature-based intrusion detection systems,” in IEEE Colombion Conference on Communications and Computing (COLCOM), 2019, pp. 1–6.
[9] T. Pietraszek, “Using adaptive alert classification to reduce false positives in intrusion detection,” in Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2004, pp. 102–124.
[10] N. Hubballi and V. Suryanarayanan, “False alarm minimization techniques in signature-based intrusion detection systems: A survey,” Computer Communications, vol. 49, pp. 1–17, 2014.
[11] D. Lin, L. Sun, K. Toh, J. Zhang, and Z. Lin, “Biomedical image classification based on a cascade of an svm with a reject option and subspace analysis,” Computers in Biology and Medicine, vol. 96, pp. 128–140, 2018.
[12] M. H. Waseem, M. S. A. Nadeem, A. Abbas, A. Shaheen, W. Aziz, A. Anjum, U. Manzoor, M. A. Balhaid, and S. Shin, “On the feature selection methods and reject option classifiers for robust cancer prediction,” IEEE Access, vol. 7, pp. 141 072–141 082, 2019.
[13] “Snort website,” https://snort.org/, 2022.
[14] “Snort rules,” https://www.snort.org/downloads/frule-downloads, 2022.
[15] P. Sun, L. Wang, and Q. Xia, “The keyword extraction of chinese medical web page based on wtf-idf algorithm,” in International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017, pp. 193–198.
[16] 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), 2019, pp. 124–129.
[17] “NVD website,” https://nvd.nist.gov/vuln, 2022.
[18] “Red Hat cve database,” https://access.redhat.com/security/, 2022.
[19] J. Nothman, H. Qin, and R. Yurchak, “Stop word lists in free open-source software packages,” in Workshop for NLP Open Source Software (NLP-OSS), 2018, pp. 7–12.
[20] M. S. A. Nadeem, J.-D. Zucker, and B. Hanczar, “Accuracy-rejection curves (ars) for comparing classification methods with a reject option,” in International Workshop on Machine Learning in Systems Biology, set. Proceedings of Machine Learning Research, vol. 8, 2009, pp. 65–81.
[21] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 6402–6413.
[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” vol. 16, no. 1, pp. 321–357, Jun. 2002.
[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations (ICLR), Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–15.
[24] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, “Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift,” in Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 13 991–14 002.