Semantic Models for Network Intrusion Detection

Peter Bednar, Martin Sarnovsky, Pavol Halas
Department of Artificial Inteligence and Cybernetics
Technical University of Kosice
Kosice, Slovakia
{name.surname}@tuke.sk

***Abstract*—The presented paper describes the design and validation of the hierarchical intrusion detection system (IDS), which combines machine learning approach with the knowledge-based methods. As the knowledge model, we have proposed the ontology of network attacks, which allow to us decompose detection and classification of the existing types of attacks or formalize detection rules for the new types. Designed IDS was evaluated on a widely used KDD 99 dataset and compared to similar approaches.**

Keywords—ontologies, network security incidents, machine learning

I. INTRODUCTION

With the extensive usage of the information and communication technologies the number and variety of the security attacks grow. This is also reflected in the growing of budget invested by companies or public institutions into the security. In order to cope with the current situation, the new and innovative techniques are applied in order to automatize the security management [1].

Recently, we can observe two main approaches to the security of the ICT: the first approach is data-oriented, and it is based on the application of machine learning techniques to proactively achieve the best possible prediction of the new attacks [2][3][4][5]. The second approach is more user-centric and it is based on the application of knowledge modelling techniques in order to model user behavior and ICT environment [8][9][10].

The presented article tries to combine these two approaches into a single system, where the domain knowledge about the types, effects and severity of the attacks is used to decompose intrusion detection task into the classification subtasks which can be handled more efficiently with less training data. The design of the proposed intrusion detection system is symmetrical in the sense that both approaches (machine learning and knowledge based) are equal and mutually contribute to address the challenges of the detection and prevention of the security threats.

The rest of this paper is organized as follows: in the following chapter we will present hierarchical knowledge model in the form of the ontology which will be used for the decomposition of the detection problem and which will provide additional contextual information. Subsequent part describes implemented machine learning models and how these models are combined with the knowledge in the ontology. Subsequent section then presents the experimental evaluation of the proposed combined approach. In this chapter we at first define quantitative evaluation metrics and then summarize the performance of the system on the standard benchmark dataset from the KDD Cup competition.

II. HIERARCHICAL KNOWLEDGE-BASED INTRUSION DETECTION SYSTEM

A. Overall system architecture

The main objective of the proposed architecture is to hierarchically decompose detection and classification of the intrusions according to the types of the attacks. For the decomposition we have proposed the Network Intrusion Ontology which main part is formalized as the taxonomy of attack types. This ontology allows to capture all knowledge related to the known types of the attacks, including the description of rare cases which are difficult to detect using the machine learning methods.

The main decomposition of the detection and classification process can be divided into the following phases:

1. Coarse attack/normal classification - this phase is implemented using the machine learning algorithm which distinguish normal traffic and attacks. If a network connection is labelled as a normal one, then an alarm is not raised. Otherwise, the suspicious connection is processed by a set of models to determine the class of attack during the phase 2.
2. Attack class and type prediction—this phase is guided by the taxonomy of the attacks from the Network Intrusion Ontology. The system hierarchically processes the taxonomy and selects the appropriate model to classify the instance on a particular level of a class hierarchy. The model can be a machine learning model statistically inferred from the training data, or rule-based model formalized using the classes and relations from the ontology.
3. When a class of attack is predicted, ontology is queried for all relevant sub-types of the attack type and to retrieve the suitable model to predict the particular sub-type. Knowledge model can also be used to extract specific domain-related information as a new attribute, which could be used either to improve the classifier's performance or to provide context, domain-specific information which could complement the predictive model.

The details about the predictive models and their evaluation will be presented in the subsequent chapter.

B. Network Intrusion Ontology

The proposed knowledge model captures all essentials concepts required to describe network intrusion systems. We have designed our semantic model according to the methodology proposed by Grüninger and Fox and with some extensions from Methodology.

The designed ontology is formalized using the OWL 2 RL profile, which allows to formalize common constructs such as multiple hierarchies and at the same time provides compatibility with the rule languages for automatic reasoning. As the objective of the knowledge model was to use it in the data analytical tasks, the concepts and properties map directly to the data used in the process. Moreover, ontology was extended with the concepts related to the classification models, to create the relation between the particular classifier and its usability on the specific level of target attribute hierarchy. The main classes of ontology include:

- Connections - This class represents the status of each connection record. It specifies Attack connection or normal traffic. Attack connections are further conceptualized using the Attack hierarchy described below.
- Effects - This class contains subclasses that represent all possible consequences of individual attacks (e.g., slows down server response, execute commands as root, etc.).
- Mechanisms - The subclasses represent all possible causes of individual ontology attacks (poor environment sanitation, misconfiguration, etc.).
- Flags - The subclasses represent normal or error states of individual connections (Established, responder aborted, Connection attempt was rejected, etc.). Each of these subclasses has a 1 equivalent instance.
- Protocols - The class contains subclasses that represent the types of the communication protocols on which the connection is running (TCP, UDP, and ICMP).
- Services - The subclasses represent each type of connection service (http, telnet, etc. ...). Each of these subclasses has a 1 equivalent instance.
- Severities - This class represents the severity of the attack, its subclasses represent the severity level (weak, medium, and high).
- Targets - The subclasses represent possible targets of a given type of attack (user, network).
- Models concept covers the classification models used to predict the given target attribute.

The instances of the specified classes represent the network connections (e.g., connection records from the data set). Trained and serialized classification models are instantiated as the instances of the Model class. The models are represented as the web resources and they could be accessed by their URI property, which points to the location where the model is serialized in the system. The main concepts and relations of the ontology are represented on the Figure 1.

The central part of the proposed semantic model is the taxonomy of Attacks which are summarized in the following figure. The taxonomy was extracted from the types of the attacks described in the KDD 99 datasets. Attacks are divided into the four main groups such as DOS, R2L, U2R and PROBE. The main types of the attacks are further specified on the additional level of the hierarchy.

Fig. 1. The main concepts of the proposed sematnic model.

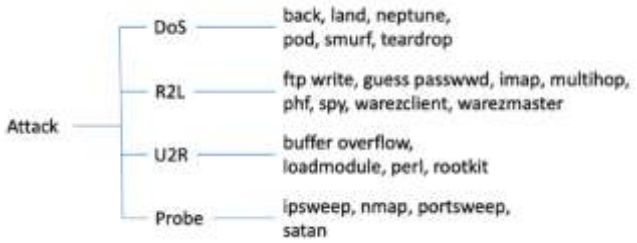

Fig. 2. The hierarchy of Attacks.

C. Machine learning models

To evaluate the proposed approach, we used the KDD Cup 1999 competition dataset, which is a commonly accepted benchmark for the intrusion detection task. The dataset consists of the records from the device logs in a LAN network collected over nine weeks. For the evaluation, we have used 10% sample with the 494,021 records in total. Each record is labeled as the normal communication or it is assigned to the major attack class and specific attack types. There are 22 different attack types which corresponds to the classes in the proposed ontology.

The common problem with the diagnostic tasks such as intrusion detection systems is that the target attribute (i.e. in our case type of the attack) is highly unbalanced with the majority of normal communication. Table I presents the taxonomy of attack types together with the number of cases in the dataset. Some attack classes such as Probe are more balanced but generally for each attack class we can find some minor types with only the few training examples. The lack of cases is

problematic not only for the training of statistical models but also for the evaluation. On the other side, rare cases can be still very critical and can in overall a big impact on the security of the system.

TABLE I. ATTACK TYPES AND NUMBER OF SAMPLES

Attack	Attack class	# of samples
back	DoS	2203
land		21
neptune		107,201
pod		264
smurf		280,790
teardrop		979
satan	Probe	1589
ipsweap		1247
nmap		231
portsweep		1040
guess_passwd	R2L	53
ftp_write		8
imap		12
phf		4
multihop		7
warezmaster		20
warezclient		1020
spy		2
buffer_overflow	U2R	30
loadmodule		9
perl		3
rootkit		10
normal	Normal	97,227

The records for each connection are described by set of features, which are represented in the ontology as the data attributes. The features can be divided into the basic features, content features and traffic features. Overall there are 32 features. The first group describes the type of the communication protocol, duration of the connection, service on the destination network node and other standard attributes describing the TCP connection. Content features are attributes that can be linked to the domain specific knowledge depending on the applications and environment in which communication occurs. The last group of features (traffic) describe the communication attributes captured during the 2 seconds time window, e.g. the number of hosts communicating with the target host etc. For the data preprocessing, we have selected only the most relevant features for the classification which were identified in the work of [4]. The final list of features includes: service, src_bytes, dst_bytes, logged_in, num_file_creations, srv_diff_host_rate, dst_host_count, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate, srv_count, serror_rate, rerror_rate,

Since the data of diagnostic tasks are commonly highly unbalanced towards the normal cases, the proposed approach is based on the decomposition of the diagnostic classification task into the hierarchy of classifiers. At the top level of the class hierarchy, an *attack detection model* is used for the prediction to distinguish between the attack connections and normal traffic. The classifier on this level was trained on the whole dataset and target attribute was transformed to the binary indicator. The main goal of this top-level classifier is to reliably separate normal connections from the attack ones.

If the top-level model detects an attack connection, the cases are further classified by the ensemble models into the one of the four types of the attack on the second level of the taxonomy (DoS, R2L, U2R, Probe). In this level, we use ensemble classifier with voting scheme trained on all attack instances (i.e. without the normal communication cases). We found that the proposed ensemble model is more efficient in the case of unbalanced target classes. The standard machine learning models proposed in the previous works were able to gain good accuracy, achieved mostly on the dominant class (in our case on KDD 99 dataset, on the most common DoS attack). However, the simple models struggled to predict minor classes such as U2R, which can be even more serious from the point of view of network security. For example, when training a decision tree model, the model has very good performance for the DoS and R2L classes but missed a significant amount of the Probe attacks and was not able to detect the U2R class at all.

Proposed weighting schema is based on the idea of complementing classifiers which is based on the performance of a particular model on the particular class. This weighting schema is presented on the Table II. The $w_{i,j}$ terms represent the weight associated with the *i*-th model and *j*-th class.

TABLE II. WEIGHTING SCHEME OF THE ENSEMBLE MODEL

Model	DoS	R2L	U2R	Probe
model 1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	$w_{4,1}$
model 2	$w_{1,2}$	$w_{2,2}$	$w_{3,2}$	$w_{4,2}$
model 3	$w_{1,3}$	$w_{2,3}$	$w_{3,3}$	$w_{4,3}$
...	...	...	...	...

After the binary classification and classification of the attack class by ensemble weighted classifier, we have trained particular models to further classify specific type of the attack on the most specific level of the taxonomy. Four different models were trained using only the records of particular attack classes (i.e. models for DoS, R2L, U2R and Probe). The most problematic was minority U2R class, as the dataset contains very few records of that type. The final implemented classification schema is presented on the Figure 3. All models were implemented in the Python environment using the standard pandas and scikit-learn

stack. Predictive models were then persistently stored and the models URIs (Uniform Resource Locators) were added as the data properties to the knowledge model.

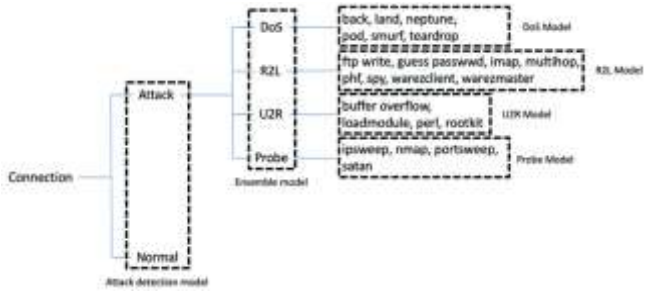

Fig. 3. The implemented hierarchical classification schema.

The main role of the semantic model in the proposed detection system is to navigate through the target class taxonomy and decompose classification problem to the sub-problems implemented by the particular models for the specific type of attack. The system is implemented using the Python language and RDFlib package which provides integration with the ontology using the SPARQL query interface. When predicting the unknown connection, system query the ontology using the SPARQL query and retrieve correspondent model for the particular class of the attacks according to the URL stored in the hasTargetAttribute property. Once the classification of the main type is performed, the system checks in the ontology if there is a classifier able to process the record further and to detect subtype of the attack.

Besides the hierarchical decomposition of the detection process, knowledge model provides also additional context which can be leveraged during the classification and improve detection of the minor classes. We have mainly extended the context with the potential effect of the attack. Additionally, if the models are not reliable enough to predict the concrete attack sub-type, the system can be used to classify attacks at least according to the severity which is retrieved from the knowledge model for the particular main class of the attack. This could serve as a supporting source of information, completing the attach type classification.

III. Evaluation

For the evaluation, we used the most common metrics employed in the classification tasks such as recall and precision. We have also computed confusion matrix for the particular classes of attacks. The confusion matrices were especially informative since they record number of correctly and incorrectly classified examples and also the types of the error. For the binary classification on the top level of the taxonomy hierarchy we used standard evaluation metrics:

- Precision: P = TP / (TP + FP)
- Recall: R = TP / (TP + FN)

where TP, TN, FP, FN are numbers of true positive, true negative, false positive and false negative records (e.g. for true positive number of records when the predicted attack was in fact attack, false positive when the predicted attack was in fact a normal traffic, false negative when the predicted normal traffic was in fact an attack, etc. The entire system was also evaluated with the number of missed attacks and raised false alarms as FAR metric (False Alarm Rate), which corresponds to the false positive records divided by total number of normal traffic records (true negative + false positive).

For the evaluation of the binary classification on the top level of the taxonomy, we used directly precision and recall metrics. In the subsequent stages on the more specific levels of taxonomy we have computed precision and recall for each class and used macro-averaging for overall evaluation. Additionally, we have computed multi-class confusion matrix to further investigate the types of the errors produced by the system.

A. Training and evaluation

For the binary classification for the attack detection, we used the decision tree classifier. Dataset includes all records and target attribute was transformed to binary indicator attack/normal traffic. The classifier was trained without the limit for maximum depth with default settings for pruning and gini index as the splitting criterion. We split the dataset randomly to 70/30 training/testing ration. The testing data were also used for overall evaluation of the entire system. Model for the binary classification achieved the accuracy 0.9997. The detailed confusion matrix is presented in the Table III.

TABLE III. Performance of the binary attack classification

	Normal	**Attack**	**Precision**	**Recall**
Normal	29,095	11	0.999	0.999
Attack	35	119,066		

For the training of ensemble classifier, we have selected only the attack records from the training set. As the base classifiers we have used various configuration of the Naive Bayes and Decision Tree models. The experiments proved that the Decision Tree classifier performed well on the Probe, DoS and R2L attacks. On the other hand, for the U2R class model produces many false alarms or (depending on pruning) the model was not able to detect U2R attacks at all. For this reason, we have trained one-vs-all model just to separate U2R class. We have then combined both types of the models into the ensemble classifier. The weights of the base classifiers were computed according to the accuracies of the models on the training data. For the evaluation we have used the same 70/30 dataset split as for the binary classification, but we have further selected only the attack records (since the normal communication is filtered already by the binary classifier). In total, models were trained on 396743 records. The confusion matrix of the ensemble classifier is presented on the Table IV.

TABLE IV. Performance of the ensemble attack classification

	Probe	**U2R**	**DoS**	**R2L**	**Prec.**	**Rec.**
Probe	1279	0	1	0	0.992	0.992
U2R	0	15	0	0	1	0.882
DoS	6	0	117,385	0	0.999	0.999

R2L	4	2	0	331	0.982	1

On the most specific level of the taxonomy, each major attack class has dedicated one model for the further classification of subtypes. The performance of each model was evaluated using the precision and recall macro-averaged for each subtype. The overall performance of the models is summarized in Table V.

TABLE V. PERFORMANCE OF THE SUBTYPE CLASSIFICATION

	Probe	**U2R**	**DoS**	**R2L**
Accuracy	0.991	0.937	0.999	0.989
Precision	0.989	0.927	0.999	0.879
Recall	0.989	0.875	0.999	0.833

The overall system with the hierarchical classification was evaluated using the standard precision, recall F-measure and FAR (False Alarm Rate) metrics. Comparison of the proposed system and models published in previous works [4][6][7][11] is presented in Table VI.

TABLE VI. OVERALL PREFORMANCE OF THE SYSTEM

Classifier	**Acc.**	**Prec.**	**F1**	**FAR**
C4.5	0.969	0.947	0.970	0.005
Random forests	0.964	0.998	0.986	0.025
Forest PA	0.975	0.998	0.998	0.002
Ensemble model	0.976	0.998	0.998	0.001
Our approach	**0.998**	0.998	0.998	0.001

Additionally, we have computed confusion matrix, which summarizes the performance for each attack class. The confusion matrix is presented in the Table VII.

TABLE VII. CONFUSION MATRIX FOR THE OVERALL PREFORMANCE OF THE SYSTEM

	Probe	U2R	DoS	R2L	Normal
Probe	1176	0	5	0	7
U2R	0	15	0	0	5
DoS	4	0	117547	0	1
R2L	3	1	0	346	7
Normal	1	0	3	1	48454

Besides the classification of attack types, we have implemented and also evaluated the classification of the attack severity. To train the severity detector we have used 10 % of KDD 99 dataset with the 70/30 training/testing ratio. The severity classifier was applied complementary to the ensemble model for the detection of the attack class. Overall achieved performance was 0.999 precision and recall with very good accuracy for the high and low severity. The Table VIII presents the confusion matrix for the severity detection in comparison for each class of the attack.

TABLE VIII. CONFUSION MATRIX FOR THE SEVERITY DETECTION

	High	Low	Medium	Prec.	Recall
DoS	117695	0	0	0.999	0.999
Probe	443	0	779		
R2L	0	346	6		
U2R	0	0	20		

Medium severity was biased by our model towards the high severity which has the similar effect like the higher false positive rate. Further details and information about the designed model were published in [9].

IV. CONCLUSION AND FUTURE WORK

In this paper we have proposed an approach based on the combination of knowledge based and machine learning methods for intrusion detection. The proposed knowledge model in the form of the ontology is used for the hierarchical decomposition of the detection process according to the types of the attack. This decomposition allows to overcome the problems with the unbalanced training data which are typical for the diagnostic machine learning tasks. By the leveraging of the domain knowledge, our combined approach also provides an additional context which includes for example the effects and severity of the attacks.

The performance of the proposed IDS is 0.998 in terms of precision as well as recall and 0.001 in terms of FAR metric, which on the standard benchmark dataset outperforms other state-of-the-art methods. Moreover, the proposed method has also potential to partially detect new emerging types of attacks in terms of the contextual information stored in the knowledge model.

In the future work we plan to extend the role of the knowledge model by introducing a rule-based classifier which will be based on the declarative rules and application of automatic reasoning technique and logical programming. We hope that this will allow to further improve accuracy for minor classes with the low number of training examples. Additionally, extended knowledge model will allow to create formalized knowledge base of the existing cases.

ACKNOWLEDGMENT

This work was supported by Slovak Research and Development Agency under the contract No. APVV-16-0213 and by the VEGA project under grant No. 1/0493/16.

REFERENCES

[1] Park, J. Advances in Future Internet and the Industrial Internet of Things. *Symmetry* 2019, *11*, 244.

[2] Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A Deep Learning Approach for Network Intrusion Detection System. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3-5 December 2016.

[3] Khan, M.A.; Karim, M.d.R.; Kim, Y. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network. Symmetry 2019, 11, 583.

[4] Zhou, Y.; Cheng, G; Jiang, S.; dai, M. An efficient detection system based on feature selection and ensemble classifier. arXiv 2019, arXiv:190401352

[5] Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J. Comput. Sci. 2018, 25, 152–160.

[6] Sharma, N.; Mukherjee, S. A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS. *Procedia Technol*. 2012, *6*, 913–921.

[7] Ahmim, A.; Ghoualmi Zine, N. A new hierarchical intrusion detection system based on a binary tree of classifiers. Inf. Comput. Secur. 2015, 23, 31–57.

[8] Abdoli, F.; Kahani, M. Ontology-based distributed intrusion detection system. In Proceedings of the 2009 14th International CSI Computer Conference, Tehran, Iran, 20–21 October 2009; pp. 65–70.

[9] Sarnovsky, M.; Paralic, J. Hierarchical Intrusion Detection Using Machine Learning and Knowledge Model. *Symmetry* 2020, *12*, 203.

[10] More, S.; Matthews, M.; Joshi, A.; Finin, T. A Knowledge-Based Approach to Intrusion Detection Modeling. In Proceedings of the 2012 IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 24–25 May 2012; pp. 75–81.

[11] Özgür, A.; Erdem, H. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ Preprints 2016, 4, e1954v1.