NOVA IMS Information Management School

MGI

Master Degree Program in

Information Management

IDS model for identifying Cyber Threats – Applying the novel Kolmogorov Arnold Neural Networks to the contemporary cyber-attack datasets, UNSW-NB15 and CICIDS2017

Miguel Branco Agostinho Ferraz Gaspar

Master Thesis

presented as partial requirement for obtaining a Master's Degree in Information Management

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

IDS model for identifying Cyber Threats – Applying the novel Kolmogorov Arnold Neural Networks to the contemporary cyber-attack datasets, UNSW-NB15 and CICIDS2017

by

Miguel Branco Agostinho Ferraz Gaspar

Master Thesis presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Business Intelligence

Supervised by

Roberto Henriques, PhD, Nova Information Management School

July 2025

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism, any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Miguel Gaspar, 20/07/2025

ABSTRACT

The Kolmogorov Arnold Neural Network variant of neural networks is a novel discovery in the field of Deep Learning introduced earlier this year. This algorithm is being hypothesized as a strong contender for superiority in performance as well as explainability in comparison to other artificial neural networks already in use. By leveraging the existing body of literature, this thesis aims to explore the performance of this novelty in deep learning when applied to Intrusion Detection Systems. Through its application onto two up-to-date datasets, the UNSW-NB15 and CICIDS2017, this thesis will provide ground for comparison between the previous state-of-the-art models and Kolmogorov Arnold Neural Networks, effectively looking to understand if this novelty can surpass previous models within the field of deep learning applied to cybersecurity.

KEYWORDS

Kolmogorov Arnold Neural Network; Intrusion Detection Systems; Deep Learning; Performance; Cybersecurity

Sustainable Development Goals (SDG):

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
AIDS	Anomaly Intrusion Detection System(s)
BiLSTM	Bidirectional Long Short-Term Memory
BiRNN	Bidirectional Recurrent Neural Network
CNN	Convolutional Neural Network(s)
CSO	Crow Search Optimizer
CVAE	Conditional Variational Autoencoder
DC	Data Center
DL	Deep Learning
DNN	Deep Neural Network(s)
FC	Fog Computing
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
IDS	Intrusion Detection System(s)
IoT	Internet of Things
KANN	Kolmogorov-Arnold Neural Network
LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
MSO	Moth Search Optimizer
NIDS	Network Intrusion Detection System
PCA	Principal Component Analysis

RNN Recurrent Neural Network(s)

SGD Stochastic Gradient Descent

SHAP SHapley Additive exPlanations

XAI Explainable Artificial Intelligence

1. INTRODUCTION

The Internet of Things (IoT) is generally defined as the network composed of sensors capable of collecting and transmitting data. The internet of things (IoT) has seen a significant increase in its infrastructural development in recent years (Radouan Ait Mouha, 2021). This infrastructure contains an ever-growing number of sensors collecting and enabling a great flow of data into data centers. This data is leveraged to produce value by the different organizations that have access to it. The expansion in the network of IoT sensors may also have new weaknesses attributed to it (Hu et al., 2017). Specifically, there is a great concern with cyber security of the data within this field, fueling research of different methods to secure the IoT networks (Li & Liu, 2021).

Cloud based IoT networks have had many challenges attributed to the characteristics of their infrastructure. This infrastructure saw a significative growth near its edges which highlighted limitations of its centrality being solely around cloud data centers (DC). The growing levels of connectivity and distance to the edges of the network led to challenges with latency, the limited bandwidth of the network and its limitations related to real time processing of data (Hu et al., 2017). In previous years these issues have been worked on by developing and deploying Fog Computing (FC) solutions on top of the previously Cloud Based infrastructure (Arivazhagan. & Natarajan., 2020; Bonomi et al., 2014). Fog Computing extends the Cloud based model of IoT by mediating between the continuum of Cloud-Thing communication. The fog nodes provide local points of access, control, computing and communication to the user, between the data centers and their endpoints (Rashid Abdulqadir et al., 2021). Lastly, apart from the challenges mentioned above, FC helps maintain the privacy of the data, secure distribution based on target users, transparency and elasticity across the network (Puliafito et al., 2019). The FC network architecture enables the deployment of various services and applications closer to the endpoints where data is collected (Radouan Ait Mouha, 2021). A trend that has surfaced in recent years is the application of machine learning algorithms to these nodes to leverage more resources in order to monitor networks more efficiently (Samann et al., 2021).

Machine learning algorithms may be used to build models that can identify and classify observable behaviors for a variety of use cases, such is the case for cybersecurity. These models

can predict events and support decision-making (Singh et al., 2023). ML plays a crucial role in securing data at the network edge against cyber threats. A key application is Intrusion Detection Systems (IDS), which use ML to identify cyberattacks (H. Liu & Lang, 2019; Saranya et al., 2020; Muneer et al., 2024). Specifically, on IoT networks, DSs deployed at fog nodes offer enhanced monitoring near IoT sensors and help protect personal data (Sahar et al., 2021).

Within the field of machine learning applied to IDS, researchers have been applying deep learning algorithms (DL) to deliver higher performance systems which can learn deeper underlying patterns between features (Lansky et al., 2021).In comparison to other machine learning models (shallow), deep learning models contain more complex structures with a higher number of learnable parameters (H. Liu & Lang, 2019). The higher number of parameters and other properties allows them to produce more accurate and complex representations of the data by sacrificing the interpretability of their outputs. This concept is usually referred to as a black-box effect (Muneer et al., 2024). Researchers, such as Muneer and colleagues in 2024, have pointed towards the importance of providing reliable explanations to the outputs of intrusion defense systems to improve their deployment and understanding, answering the question of "Why is this tagged as an intrusion?" (Muneer et al., 2024). Houda and colleagues have highlighted explainable artificial intelligence (XAI) as a rising paradigm.

There have been a lot of developments made by other researchers which include a plentitude of models utilizing different combinations of algorithms and datasets. These models have been developed to reach high accuracies of over 99% which remain competitive against many of the deep learning models. Within the sub field of deep learning, most models tend to achieve very high accuracies. To compare these models, researchers will often divert to analyzing their efficiency, speed or processing weight. Examples of the algorithms being used, which achieve relevant performances, are the recurrent neural networks and convolutional neural networks (Muneer et al., 2024). Other aspects from which to expand on the capabilities of IDS'S, besides the choice of algorithms, exist and may be considered. These may include different techniques for feature selection, which can greatly impact the output of a model (Rashid et al., 2020) both in its accuracy and speed at producing said output. In a paper from 2022, Albulayhi and colleagues highlight feature selection as a great driver for improvements on IDSs (Albulayhi et al., 2021).

Earlier this year, Liu and colleagues introduced a novel deep learning algorithm called the Kolmogorov-Arnold Neural Network (KANN) (Z. Liu et al., 2024). Unlike traditional neural networks, KANN replaces fixed weights with flexible functions, making it potentially more accurate and easier to interpret. Early research suggests that KANN could outperform common models in certain tasks (Z. Liu et al., 2024), making it a promising candidate for further exploration in areas like intrusion detection, where both accuracy and explainability are critical (Muneer et al., 2024). Additionally, being considered a novelty, the algorithm is prone to be tested in other nuances such as efficiency of training and computational weight, factors that are critical to the success of IDS deployed onto an IoT scenario (Abdullahi et al., 2022).

When training a ML model for Intrusion Detection Systems, the choice of datasets significantly influences the thresholds of accuracy a model can achieve. The list of benchmarked datasets includes KDD CUP 1999, NSL-KDD, CICIDS-2017, and UNSW-NB15, each providing various attack types and network traffic patterns. Considering the critiques about the outdated nature of datasets like KDD CUP 1999 and NSL-KDD, literature highlights the importance of contemporary and diverse datasets that contain relevant patterns for cyber threats (Khanan et al., 2024; Moustafa & Slay, 2015). In this thesis, the UNSW-NB15 and CICIDS-2017 datasets were selected for their robust and current representation of network attacks, extensive feature sets, and recognized benchmark status in the IDS research community.

So far, we briefly mentioned IoT network architecture, fog computing, cyber security, IDS and machine learning. The body of literature in this field is rich, however, with novelties being introduced frequently, gaps in literature may be outlined. The aim of this thesis is to fill a literature gap by developing a network intrusion detection system through applying the novel KANNs to the UNSW-NB15 and CICIDS2017 datasets. The choice of KANN follows the relevance of their recent introduction. The choice of datasets follows the relevance of both UNSW-NB15 and CICIDS2017 as up-to-date datasets containing contemporary patterns of network traffic data. To achieve a final model, the CRISP-DM framework will be utilized, allowing for the inclusion of the considerations to do with feature selection as central problem in IDSs.

Summarizing the proposed aims of this thesis:

1. This thesis will start by providing a critical literature review over the application of deep learning to IDS

2. Proposal of a methodology to develop a KANN based NIDS model applied to the UNSW-NB15 and CICIDS2017 dataset
3. Evaluation of different data processing regarding feature selection and resampling techniques
4. A discussion comparing the best performing KAN based model AG the performance of other

   models present in literature

2. LITERATURE REVIEW

2.1. TAXONOMY OF INTRUSION DETECTION SYSTEMS

Intrusion Detection Systems are common security systems used for the protection of infrastructures and computers against malicious activity. IDSs capture and analyze the network traffic to detect suspicious activity and enable the network administrators to take measures to secure their assets (Axelsson, 2000; Kumar & Singh, 2020). Axelsson (2000, p. 2) as referred to them as being the burglar alarms for the field of computer security. Intrusion Detection Systems have been studied for multiple decades, having the first concrete model been introduced by Denning (1987) as an application which aims at detecting a large range of security breaches within a domain. Since then, many authors have helped grow our comprehension of what are the different characteristics of an IDS to consider, resulting in a complex taxonomy. The complex taxonomy allows for the segmentation of different models developed so far, with different strengths (Mukhaini et al., 2024).

2.2. DEPLOYMENT STRATEGIES

It is essential to understand how these systems are deployed across network infrastructures. The deployment strategy, whether host-based or network-based, determines where data is collected. This section examines the primary deployment classifications proposed in the literature.

Axelson (2000) introduced a simple taxonomy for IDS, classifying them on basis of their detection techniques. In 2019, Liu & Lang (H. Liu & Lang, 2019) reviewed and proposed a taxonomy for IDSs dividing them on the basis of the detection technique they utilize and the sources of the data used (H. Liu & Lang, 2019). Within the data sources, the authors considered both host-based and network-based methods. Host-based models work with data that is within the host into which the model is deployed. A downside of host-based models is that the hosts usually have low availability of resources (IoT sensors), and the models are limited in terms of complexity. The model works with the logs produced by the operating system it is deployed on (H. Liu & Lang, 2019). Network-based models allow for the deployment of resource-heavy models onto hosts, focusing on monitoring over the flow of network traffic data in a segment of the network. (H. Liu & Lang, 2019). Other authors have referred to this classification based on

sources of data as “deployment method” classification (Z. Ahmad et al., 2021, p. 7) or architecture classification (Momand et al., 2023, p. 6).

2.3. DETECTION TECHNIQUE

A core component in defining IDS lies in the detection strategy adopted. Whether an IDS identifies known threats through pattern matching or detects new attacks through discerning behavior analysis significantly affects its architecture and applicability. This section presents the main detection paradigms that are identified in literature.

Liu & Lang (2019b) also consider detection technique as a classification term, dividing the models into anomaly-detection or signature-detection, in line with what had been proposed by Axelson (2000). This binary division is supported by other researchers such as R. Ahmad & Alsmadi (2021), which emphasize the importance of deep learning in effectively identifying a significant taxonomy of attack types, as well as Ahmad and colleagues (2021). Signature-detection models work on pattern matching the incoming instances with attack-type instances, learnt during training (H. Liu & Lang, 2019). While the signature-detection model will compare incoming traffic with what is known as an anomaly to secure the network, anomaly-based detection models will learn the normal behavior patterns of the network during training and flag any incoming unrecognized instance. The model proposed by Denning (Denning, 1987) uses statistical inference to identify abnormal behavior, this model could be classified as an anomaly detection model in the present. Anomaly detection-based models may be more effective at protecting the network against unseen attack types, while signature-based models allow for the precision at identifying which known attack is present in an instance (H. Liu & Lang, 2019).

In their literature review, Albulayhi and colleagues (2021) explore the field of IDS through the scope of the different deep learning algorithms that have been developed. The authors introduce a taxonomy for the attack types to be considered by IDS as well as provide a simple taxonomy categorizing IDSs based on their detection technique (Albulayhi et al., 2021). The authors expand the two categories mentioned by Liu & Lang (2019) and Axelson (2000) into four, now considering both hybrid-detection models and specification-based models (Albulayhi et al., 2021). Hybrid models use a combination of anomaly and signature-based detection, usually composed of heterogeneous algorithms. By joining together both signature and anomaly-detection (Albulayhi et al., 2021), this enables the hybrid model to aggregate the benefits of both types of detection techniques, allowing it to detect both known and unknown attack types

(Otoum & Nayak, 2021, p. 10). Specification-based models are also considered by Albulayhi and colleagues (2021) and are defined by other researchers as the detection of abnormal behavior in a network based on a hard set of predefined rules, verifying the execution of a protocol (A. Le et al., 2016). This detection technique is compared to anomaly-detection since it counts heavily on domain knowledge to outline the abnormal behavior rules (Jamalipour & Murali, 2022). Other researchers have disregarded hybrid IDSs and considered only anomaly, signature and specification-based approaches as categories for classification (Pundir et al., 2020).

Najafli and colleagues (2024) produced a systematic review of the literature on deep learning techniques applied to IDSs where they considered a taxonomy which leverages similar ideas from previous studies. The authors used a different terminology to refer to the detection-technique dimension, calling it analysis strategy, and, unlike a portion of the recent studies (Albulayhi et al., 2021; Jamalipour & Murali, 2022), solely considered anomaly and signature-based techniques as part of their classification (Najafli et al., 2024). The authors chose to refer to the dimension of detection strategies by the deep learning modes (discriminative and generative, for example) used, as well as the classification type (binary and multi-class, for example) performed (Najafli et al., 2024).

Arisdakessian et al. (2023) touch on a paradigm of classifying IDS's that integrates traditional detection techniques (anomaly signature detection) as well as specification-based, which

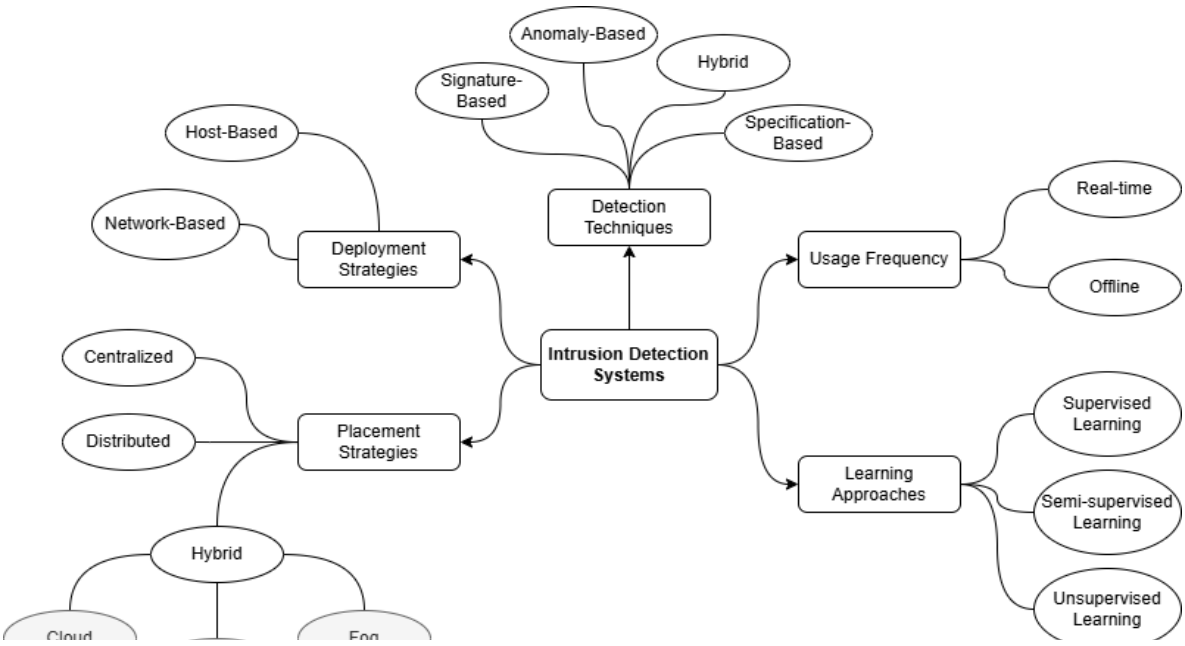

Figure 2.1 - Original Visual Representation synthetizing the lifted taxonomy for IDS

parallels the analysis strategies discussed by Pundir and colleagues (2020). Arisdakessian et al. (2023) provides a more holistic framework by placing these detection techniques within a higher-level taxonomy, such as trust-based and data-driven models. This broader view allows the authors to address emerging challenges in IoT environments by combining methodologies, thus expanding on the binary detection strategies discussed by Liu & Lang (2019) and Najafli et al. (Najafli et al., 2024). Furthermore, while Najafli et al. focus on deep learning modes and platform-specific deployments, Arisdakessian et al. (2023) emphasize the role of trust and mathematical models, enriching the taxonomy with interdisciplinary concepts such as game theory and blockchain (Arisdakessian et al., 2023).

2.4. PLACEMENT STRATEGY

As IoT and fog computing become more widespread, placement strategies must consider distributed resources and decentralized environments. This section reviews the taxonomy around IDS placement and explores emerging perspectives on platform-specific implementations.

Jamalipour & Murali (2022) expanded on the aforementioned studies by considering a new dimension of IDSs. In their study, the authors mention the placement strategy implemented in the deployment of an IDS model as a dimension, parallel to detection techniques and data sources/deployment strategy (Jamalipour & Murali, 2022; Mukhaini et al., 2024). This dimension contains three categories This dimension includes three categories: centralized, where the IDS is positioned at a central node for monitoring; distributed, where each node in the network has its own IDS implementation for local detection; and hybrid, which combines aspects of both centralized and distributed approaches for optimized resource use and faster attack detection (Jamalipour & Murali, 2022). These authors were also referenced by Mukhaini and colleagues in 2024 in their systematic review, considering the same taxonomy with the added dimension for placement strategies (Mukhaini et al., 2024).

Within the placement strategy, mentioned by previous authors (Z. Ahmad et al., 2021; Momand et al., 2023) Najafli and colleagues expanded the dimension, providing a new branch for the platform chosen for placement (Najafli et al., 2024), besides the architecture used (Momand et al., 2023). The authors provide a layer of consideration for the platforms of deployment of IDSs into the contemporary characteristics of an IoT network with different processing platforms, fog, cloud and local computing (Najafli et al., 2024).

Arisdakessian et al. (2023) complement the existing classifications by proposing a new high level approach-based taxonomy while reinforcing the importance of placement strategies previously discussed (Jamalipour & Murali, 2022; Mukhaini et al., 2024). Their focus on centralized, distributed, and hybrid deployments aligns with previous studies

2.5. USAGE FREQUENCY

IDSs can operate in real-time (online) or process data retrospectively (offline), and this usage frequency impacts their responsiveness and resource demands. This section introduces the concept of usage frequency as an emerging classification dimension and highlights its relevance in contemporary IDS design and evaluation.

Kamaldeep et al. (2020) provides a taxonomy which has its scope mixed between specific categories of IDSs and characteristics on IDS' related studies. When focusing on categories that relate specifically to IDSs, the authors introduce a taxonomy which stays in dialogue with most of the already reviewed studies, however, introducing a new dimension, usage frequency (Kamaldeep et al., 2020) . Usage frequency refers to how often the IDS performs detection tasks, distinguishing between real-time detection from data streams (on-line) (Alsaleh et al., 2024), where monitoring happens continuously as data flows, and offline detection, where analysis is done in batches at intervals (Kamaldeep et al., 2020).Therefore, the authors add depth to traditional classifications and offer more nuanced criteria to evaluate and compare IDSs.

2.6. LEARNING APPROACHES

Within the field of AI and its subfields, the learning processes applied to the development of IDSs, are commonly categorized into three approaches: supervised learning, unsupervised learning, and semi-supervised learning. Each of these approaches encompasses distinct algorithms that leverage different structures and methods to enhance the efficiency and accuracy of IDS (Albulayhi et al., 2021; Saranya et al., 2020).

Supervised learning algorithms deal with class labeled data and their process of learning includes training to find the relationship between the input features and the class labels. Some of the common supervised learning algorithms applied to IDSs include Support Vector Machines (SVM), Neural Networks, Ensemble Methods, Logistic Regression (LR), and Decision Trees (DT). Within this learning methods problems may be defined into either classification or regression,

for categorical class outputs or continuous class outputs, respectively (Saranya et al., 2020). Samann et al. (2021) concluded the prevalence of adoption of supervised classification models for IDS development.

Unsupervised learning algorithms do not require training and work to find the hidden structure of unlabeled data (Saranya et al., 2020). Within unsupervised learning algorithms there are three main branches. Clustering algorithms such as k-means and k-mediods, association analysis with association rules and dimensionality reduction with principal component analysis (PCA) (Saranya et al., 2020) or auto-encoders (AE) with an established reconstruction error threshold (Albulayhi et al., 2021).

Semi-supervised learning algorithms use both labeled and unlabeled data, combining elements of supervised and unsupervised learning. These algorithms are particularly useful in situations where labeling large datasets is impractical (Saranya et al., 2020). Typically, they are trained on normal traffic patterns, where a small portion of the data is labeled, and the rest is unlabeled, helping to detect anomalies or deviations from expected behavior (R. Ahmad et al., 2023). Unsupervised approaches, such as Generative Adversarial Networks (GANs), can be adapted for semi-supervised learning, working to identify deviations from normal network behavior (Albulayhi et al., 2021).

Kamaldeep et al. (2020) focused solely on two learning approaches, supervised and unsupervised learning. Similarly, other researchers have narrowed the scope of their work to concentrate exclusively on these learning approaches in relation to specific detection techniques, as is the case with Ahmad et al. (2023). In their paper, Ahmad et al. explore anomaly-detection models using all three learning approaches, supervised, unsupervised, and semi-supervised. The authors emphasize the use of each approach based on the availability of labeled data and the need to handle zero-day attacks (R. Ahmad et al., 2023). Otoum & Nayak (2021) conducted a study that applied both unsupervised and supervised learning to signature-based and anomaly-based detection techniques. In contrast, other studies have developed hybrid IDSs using only one learning approach, such as Bostani & Sheikhan (2017), who focused exclusively on unsupervised methods for anomaly detection.

Multiple authors have contributed to expanding our comprehension of characteristics with which we may define an IDS model. Most studies agree on the definitions of at least three types of IDS models depending on their detection techniques being hybrid, signature and anomaly

detection. The way in which the models employ different detection techniques is closely tied to the learning approaches used. Furthermore, literature also converges on the binary definition for the data sources, classifying IDS into being either host-based or network-based.

2.7. REVIEW OF DEEP-LEARNING STUDIES ON IDS

To delve into the specific work done on different algorithms, a brief literature review was conducted. The list of selected works to be considered was obtained from previously referenced literature reviews (Issa et al., 2024; Momand et al., 2023; Mukhaini et al., 2024) and enriched by considering novel works published in the past 3 years, extracted from Scopus and Google Scholar. This section focuses on the proposed models presented in the following research papers.

Table 2-1 - Literature Review table Summarizing various works on DL Models for IDS

Author	Year of Publication	Algorithms	Dataset	Metrics obtained
K.-H. Le et al.	2022	CNN	UNSW-NB15 CICIDS2017	96,69% Accuracy (UNSW-NB15) 97,22% F1-Score (UNSW-NB15) 95,92% Accuracy (CICIDS2017) 96,34% F1-Score (CICIDS2017)

Zhao et al.	2022	CNN	UNSW-NB15 BOT-IoT	86,11% Accuracy (UNSW-NB15) 87,02% F1-Score (UNSW-NB15)
Ullah & Mahmoud	2022	CNN-BiLSTM	BoT-IotT MQTT IOT-DS2 NSLKDD	99.82% Accuracy (NSL-KDD) 99.48% Accuracy (IOT-DS2)
Keserwani et al.	2023	GA-DNN	UNSW-NB15	98,1% Accuracy
Mahmood & Al Dabagh	2023	DNN	MQTTS	95.87% Accuracy on Balanced with ADAM 94.40% F1 on Balanced with SGD

El-Rady et al.	2023	CNN	CSE-CICIDS2018 UNSW-NB15	99,18% Accuracy (CICIDS2018) 98,01% F1-Score (CICIDS2018) 97,39% Accuracy (UNSW-NB15) 97,51% F1-Score (UNSW-NB15)
Muruguesh & Murugan	2023	CRNN	WSN-DS	99,72% Accuracy 99,45% F1-Score
Syed et al.	2023	CNN-BiLSTM	BoT-IoT	99,55% Accuracy 99,49% F1-Score
Henry et al.	2023	GRU CNN	CICIDS-2017	98,73% Accuracy

Kasongo	2023	RNN	UNSW-NB15 NSL-KDD	74.19% Accuracy (UNSW-NB15)
Sharma et al.	2023	DNN	UNSW-NB15	90,9% Accuracy 91% F1-Score
Sajid et al.	2024	XGBoost-GRU	UNSW-NB15 NSL-KDD WSN-DS CICIDS-2017	90,72% Accuracy (UNSW-NB15) 90,72% Accuracy (NSL-KDD)

Amouri et al.	2024	KANN	N-BaIoT	99,69% Accuracy 98,04% F1-Score

Keserwani et al. (2023) developed a Network Intrusion Detection System (NIDS) using the UNSW-NB15 dataset. They applied a genetic algorithm for feature selection and designed a Deep Neural Network (DNN) with four hidden layers for the classification task. To mitigate overfitting, they incorporated dropout as a regularization technique, achieving an accuracy of 98.11% on the UNSW-NB15 dataset. However, one critique of their approach is the focus on binary classification, which could have been extended to the more challenging multi-class classification scenario. In a similar study, Mahmood & Al Dabagh (Mahmood & Al Dabagh, 2023) developed an Intrusion Detection System (IDS) using a DNN, with an emphasis on creating a lightweight solution. In their work, the authors developed an IDS model on top of a specialized dataset, MQTT. The feature selection technique used consisted of removing the features with only null values and string format data (Mahmood & Al Dabagh, 2023). The resulting input dataset consisted of 19 features. Although feature selection was performed, there is no mention of other feature selection techniques being used that could be applied to more complex datasets. Like Keserwani et al. (2023), they utilized ReLU activation functions in the hidden layers of their network (Mahmood & Al Dabagh, 2023). Additionally, a limitation of their study is the choice of the MQTTset dataset, which is specialized and makes comparative analysis with general attack datasets more difficult (Khanan et al., 2024).

Maithem & Al-Sultany (2021) developed a deep neural network (DNN) model using the KDD-99 dataset. The model described topology consisted of an input layer with 125 neurons, followed by three hidden layers with 50, 30, and 2 neurons, respectively, and an output layer with 5 neurons for multi-class classification. The authors employed *ReLU* activation functions in the hidden layers and a *Softmax* activation function in the output layer. The hyperparameters, including the choice of the final optimizer (ADAM), were fine-tuned during training. The model achieved an accuracy of 99.98% on the test data in a multi-class classification scenario (Maithem & Al-Sultany, 2021).

In their study, Y. Yang et al. (2019) developed an intrusion detection model using a DNN for classification. This model leveraged a conditional variational autoencoder (CVAE) to learn the probabilistic distribution of the input data. The trained CVAE parameters were then used to initialize the weights of DNN before training. The CVAE also enabled oversampling of the original datasets by generating new samples from the known distributions, which were added to the DNN's training data (Y. Yang et al., 2019). The authors compared this initialization and oversampling approach on the NSL-KDD and UNSW-NB15 datasets against other techniques like SMOTE, ADASYN, and ROS. The results showed that their method outperformed other oversampling approaches in terms of accuracy and f1-score (Y. Yang et al., 2019).

Houda et al. (2022) proposed a DNN-based IDS with an architecture consisting of five hidden layers using Leaky Rectified Linear Unit (Leaky ReLU) activation functions, offering a different choice of activation function compared to those used in previous studies (Mahmood & Al Dabagh, 2023; Maithem & Al-Sultany, 2021; Yang et al., 2019). The model included an output layer for a binary classification scenario. The datasets chosen were UNSW-NB15 and NSL-KDD and the model obtained 0.88% and 0.99% accuracy, respectively. However, the primary focus of the study was not the model's performance. Contrary to the studies mentioned previously, this study aimed at developing an explainable AI framework. By applying RuleFit, SHAP, and LIME, the authors were able to explain the model's classification decisions, thus enhancing the interpretability and empowering future users to better understand and trust the decisions made by the IDS. The authors leave a clear remark on the importance of considering explainability for future IDS model developments (Houda et al., 2022).

TabNet was introduced as a deep learning model designed for attention-based classification on tabular data (Arik & Pfister, 2021). In TabNet, neural networks are used in the form of a

Feature Transformer, which processes the input data at each step. The model operates as a sequence of decision steps, where at each step, an Attentive Transformer performs feature selection by focusing on a subset of the input features. This subset of features is processed by the Feature Transformer to extract feature representations, which are used to make a partial decision. The selected features are masked after each step, meaning that the attention mechanism focuses on different features at each step for each sample. This process repeats multiple steps, and the partial decisions from each step are combined to produce the final classification. The use of attention-driven feature selection ensures that the model dynamically adapts to the input data, allowing for efficient and interpretable predictions without the need for extensive preprocessing. This model was adapted by Nguyen et al. (2022) for deployment in an IDS classification problem. The authors observed that the model could identify known attacks while achieving 97.95% accuracy on UNSW-NB15 in a multiclass scenario. They also tested the model's ability to identify unknown attacks by omitting instances from one attack class during training and using a binary classification setup to distinguish between known and unknown attacks. The model achieved 99.93% accuracy in detecting unknown attacks (Nguyen et al., 2022). However, the approach may have limitations in identifying novel attacks, as it might be learning to recognize the signature of an unlabeled attack type rather than detecting fundamentally different attack patterns. Although TabNet developers Arik & Pfister (2021) touched upon the factor of explainability within the outputs of the attention mechanisms of the model, this was not considered in Nguyen et al. (2022) paper.

Assy et al. proposed the use of one-dimensional CNN, in contrast to previous studies that utilized CNNs for two-dimensional classification tasks (Assy et al., 2023). To address class imbalances, the authors applied the ADASYN oversampling algorithm and provided a detailed description of their preprocessing stages. Consideration for model complexity is described in their research as well as the trials for the best optimizer for the CNN model, concluding with ADAM. The authors highlight the capabilities of CNNs to extract features replacing the need for manual feature engineering. The final model achieved an accuracy of 93.2% and 93.1% F1-Score on the NSL-KDD dataset (Assy et al., 2023).

El-Rady et al. (2023) proposed a CNN classification model that leverages the feature extraction capabilities of convolutional layers while utilizing fully connected layers for the classification task. The architecture is composed of multiple convolutional blocks, incorporating

batch normalization to optimize training times and max pooling for feature compression. Although the authors did not specify the use of any hyperparameter optimization algorithm, the model demonstrated strong performance on two datasets, achieving 99.7% accuracy on the UNSW-NB15 dataset and 99.18% accuracy on the CSE-CICIDS2018 dataset. Furthermore, the authors observed that the proposed model surpassed a benchmark random forest model.

Erza et al. (2022) proposed a lightweight CNN model based on EfficientNet by Tan & Le (2019), aiming to optimize both performance and reduce computational complexity. To further enhance efficiency, the authors utilized the random forests embedded method for feature selection, identifying the most important features before converting the data into images. The model was then trained using Stochastic Gradient Descent (SGD). The primary goal was to create a lightweight model suitable for deployment in an IoT context. However, despite these efforts, the model developed by Erza et al. (2022) remains more complex than alternatives like Chen et al.'s (2022) model, without offering a significant performance boost. It achieved an accuracy of 99.91% on the AWID2 imbalanced dataset, slightly below the 99.96% accuracy attained by Chen et al. (2022). However, Ezra et al considered the importance of applying other metrics to mitigate the bias of the accuracy metric in an imbalanced data scenario, therefore resorting to F1-Score as the primary metric. Liu et al. (2021) have also taken advantage of the full capabilities of CNN while implementing feature selection embedded methods seen in ML ensembles. In their paper, the authors utilize XGBoost feature selection, as well as random forest feature selection previously to training the CNN on the reduced dataset. The model developed by Liu et al. (2021) obtained an accuracy score of 95,15% on a privately collected dataset, and 99,89% on a publicly available dataset, BaIot (Meidan et al., 2018),over both feature selection methods. The major difference was observed to be the number of features used to achieve these metrics, which was lower for the model using XGBoost feature selection. The authors considered evaluating computation time for these models, and it was concluded to be faster than that of a K-Nearest Neighbors model approach (Z. Liu et al., 2021). As a final observation on the work developed by Z. Liu et al., the model exhibited a very poor F1-score for the multi-class problem on the CCD-INIDV1 dataset, and the authors did not provide an explicit accuracy evaluation for this dataset in a multi-class classification scenario.

Le et al proposed another lightweight CNN model, a network composed of 10 layers and 890826 parameters (K.-H. Le et al., 2022), which is considerably more than the parameters in

the model proposed by Chen et al. (2022). The model presented is developed on top of the UNSW-NB15 as well as CICIDS2017 datasets. This study also takes advantage of a generative adversarial network to minimize class imbalances in the datasets by generating attack data. Although this oversampling technique is displayed to enhance the evaluation metrics provided for the model, these results are still inferior to previously presented results (El-Rady et al., 2023), obtaining 96,69% accuracy with UNSW-NB15 and 95,92% with CICIDS2017 (K.-H. Le et al., 2022).

Zhao et al. has also proposed the application of convolutions onto a lightweight neural network to enable stronger feature extraction, proposing the combination of this technique with principal component analysis (PCA) for dimensionality reduction (Zhao et al., 2022). The authors support the choice of PCA as a lighter implementation of a linear transformation instead of using nonlinear transformations, such as the common use of auto-encoders. The authors support this preprocessing on the need for a lightweight model and, through the means employed, achieve a model with only 5082 parameters. The accuracy achieved by this model is not impressive when compared to the previously mentioned studies with an accuracy of 86,11% on the UNSW-NB15 dataset on a multiclass scenario. The main contribution of this study Is regarding the lightweight capabilities of convolutions for feature extraction and PCA for dimensionality reduction (Zhao et al., 2022).

Abdul Lateef et al. (2020) proposed a model utilizing a classic RNN algorithm. Their study aimed to test the RNN's capabilities and analyze the efficiency of using the crow swarm optimizer algorithm for feature selection, optimizing the problem on the criterion of lowest feature count and best performance in accuracy. The model achieved an accuracy of 98.34% on the KDD-99 dataset. They concluded that RNN is suitable for binary intrusion classification without exploring the multiclass scenario. Additionally, the authors concluded the positive effect of using CSO for feature selection (Abdul Lateef et al., 2020). Murugesh & Murugan (2023) developed an RNN-based model, also leveraging CNN’s feature extraction capabilities. Their proposed CRNN used 4 convolutional layers to transform the data and extract features before feeding them into the RNN, sequentially. Unlike Abdul Lateef et al. (2020), they employed the moth search optimizer (MSO) algorithm for hyperparameter tuning. Similarly to Keserwani et al. (2023) authors highlight the use of dropout layers to mitigate the existence of overfitting during training (Murugesh & Murugan, 2023). The MSODL-ID model achieved 99.72% accuracy on a specialized dataset (Murugesh & Murugan, 2023). The authors conclude the positive effect of the MSO

algorithm on the classification performance over different ratios for splitting the data, and the superiority of the proposed model against various machine learning models (Murugesh & Murugan, 2023).

Authors Al & Dener proposed a CNN-LSTM model, utilizing a combination of SMOTE and Tomek-Link for oversampling and under sampling to balance the training data (Al & Dener, 2021). They built this model using the UNSW-NB15 dataset, achieving 99.83% accuracy on a binary classification scenario (Al & Dener, 2021). Ullah & Mahmoud (2022) compared a selection of models, including one like Al & Dener's (2021) CNN-LSTM model. Their model was a CNN-BiLSTM, which incorporated the advantages of a bidirectional RNN and used SMOTE. They concluded there was a slight improvement in accuracy from CNN-LSTM to CNN-BiLSTM, while emphasizing the strengths of combining CNN, SMOTE, and RNNs in general (Ullah & Mahmoud, 2022). A similar study was carried out by Syed et al. (2023), comparing the performance of a simple RNN model with a Bi-LSTM model. The author observed greater performance and ability to consider the long-term dependencies in the data on the Bi-LSTM model (Syed et al., 2023). Additionally, Syed et al. (2023) also confirmed the importance of feature selection in the development of these models.

Henry et al. (2023) proposed an IDS model leveraging from the CNN algorithm, specifically utilizing GRU units before the output layers, after feature extraction using multiple common convolutional layers. The authors further reduced the complexity of the datasets by applying Pearson's Correlation filter method (Henry et al., 2023), contrary to the previously mentioned studies. The model was developed using the CICIDS-2017 dataset and was evaluated to achieve 98,73% accuracy with only 58% of the original features (Henry et al., 2023).

A study carried out by Kasongo (2023) on the comparative analysis of algorithms, such as GRU, LSTM, and SimpleRNN, highlighted the superiority of the LSTM models in multi-class classification, using both UNSW-NB15 and NSL-KDD datasets (Kasongo, 2023). The authors considered the scenario where computing resources might be limited and applied XGBoost as an embedded feature selection method. The authors concluded the superiority of a SimpleRNN model when evaluated against a test set where it obtained 74.19% accuracy on UNSW-NB15. The evaluation process is thoroughly described and includes the number of units used (Kasongo, 2023).

Sajid et al. (2024) developed a study which aimed at comparing the performance of feature selection using XGBoost against feature extraction using CNNs. The authors developed LSTM and GRU based models using these preprocessing techniques and a large variety of datasets, including UNSW-NB15. SMOTE was applied to mitigate class imbalance by oversampling minority classes (Sajid et al., 2024). The authors note the superiority of the XGBoost-GRU model, in comparison to other proposed combinations of LSTM, CNN and GRU, achieving slightly better performance with 90,72 accuracy on binary classification (Sajid et al., 2024). The authors conclude by criticizing their model on the basis of large training times and complexity (Sajid et al., 2024).

Sharma et al. (2023) proposed a three-hidden-layer DNN architecture-based model developed over the public dataset UNSW-NB15. To make the model more efficient, they resorted to the use of filter methods for feature selection instead of more robust methods such as XGBoost, as seen in previous studies. Additionally, they addressed class imbalance issues with GANs to generate synthetic instances for the minority class (Sharma et al., 2023). The authors performed a thorough performance evaluation by splitting the dataset into train, validation, and test sets in a 60:15:25 ratio, as well as utilizing a great variety of metrics. As a result, the final GAN-DNN model obtained a 91% F1-score in comparison to the 84% obtained by the sole DNN model in a multiclass scenario (Sharma et al., 2023). Moreover, the model was observed to perform better than a selection of other ML models, achieving 90.9% accuracy. The authors concluded that the GAN had a positive effect on the classification performance of the DNN. However, a limitation of this study may be attributed to the classification being performed on a reduced space of classes, as the classes were grouped into only 5, compared to the original 10 available in UNSW-NB15 (Sharma et al., 2023).

In august 2024 the first application of the KANN to IDS was made by Amouri and colleagues (Amouri et al., 2024). The team developed an ensemble consisting of XGBoost and KANN, falling in line with the strengths of ensembles, highlighted by previous research (Rashid, M., 2020). This model architecture deployed the novel KANN to process the data and then feed its outputs onto the XGBoost algorithm to produce the final classification. The ensemble stack was trained on the N-BaIoT dataset (Amouri et al., 2024). N-BaIoT is a publicly available specialized dataset introduced in 2018 which contains data collected from commercial IoT devices infected by a botnet (Meidan *et al*., 2018). The model developed by these researchers may be identified as a

network intrusion detection system which focuses on signature detection, due to its focus on learning the abnormal patterns in the network data provided (Amouri et al., 2024). The authors evaluated the model to achieve 99.69% accuracy and 98,04% f1-score on the N-BaIoT dataset (Amouri et al., 2024).

These studies highlight the development and evaluation of IDS models featuring different combinations of DL algorithms and architectures. Additionally, we may highlight a large variety of aspects that contribute to the success of these models, for example, optimization algorithms, preprocessing techniques and feature selection techniques. Furthermore, and derived from these aspects, various strengths and limitations can be attributed to most of these models. These strengths and limitations may be considered when performing an evaluation for new models, specifically those that include novelties, such as Kolmogorov Arnold Networks.

3. METHODOLOGY

In this section, we present the proposed framework for the development of a signature-based NIDS model, using the novel Kolmogorov Arnold Neural Networks algorithm. We will provide brief introduction and discussion into the different tools used, such as the algorithms, datasets and evaluation metrics that will be included in this work. Additionally, it may be noted that the choice of techniques we chose to compare in this thesis is justified in the recurring techniques seen usage in literature.

3.1.1. PROPOSED MODEL

The proposed model aims at filling a gap in literature by developing and evaluating a deep learning-based NIDS model which leverages the novel Kolmogorov Arnold Network algorithm and applies it to relevant network IDS datasets, such as UNSW-NB15 and CICIDS-2017.

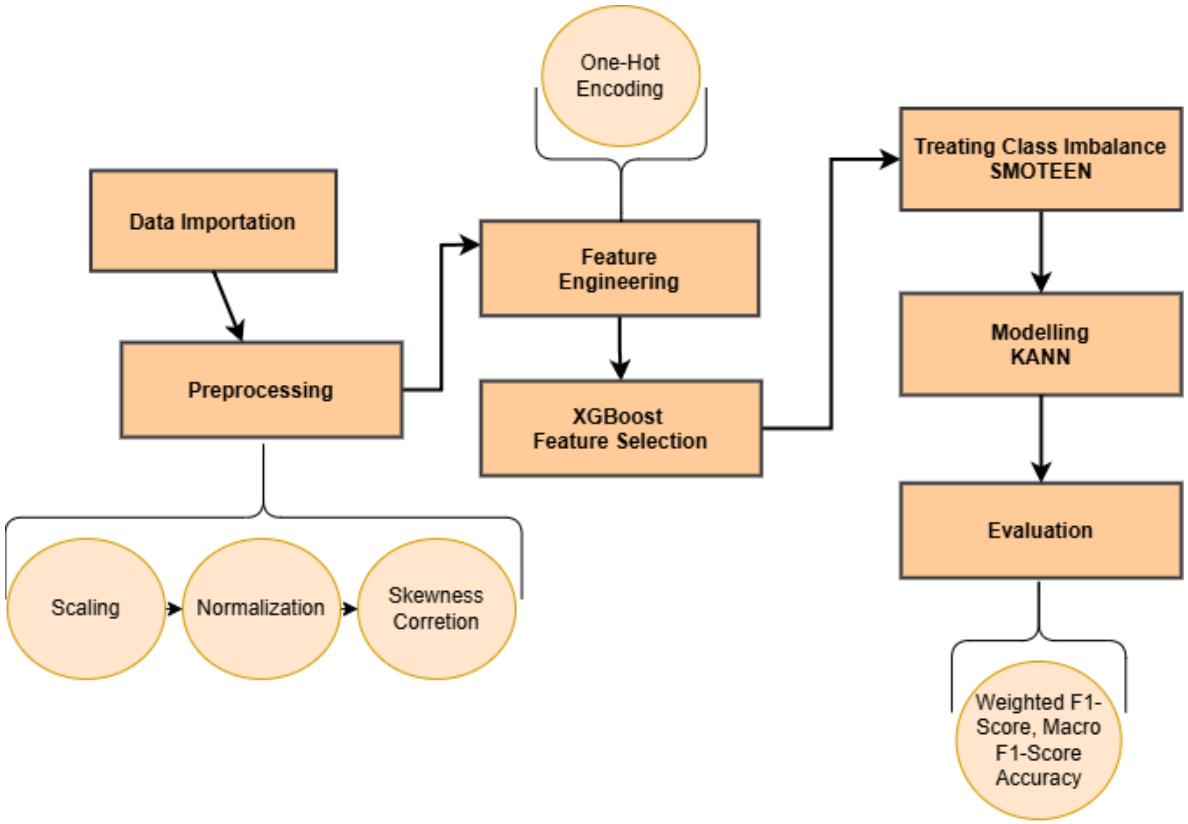

Figure 3.1 - Framework for the proposed model development process

To achieve this, data from both data sets was preprocessed, filtered using various feature selection techniques and resampled using different techniques to ensure its best suitability for the job of classification done by the KANN algorithm. A thorough display of the steps followed can be seen in *figure 3-1* and the decisions taken regarding the various choices at each step will be discussed in this section. This methodology allows the production of a competitive model using KANN that may be compared to existing state-of-the-art models. Furthermore, the code used for the development of our thesis is available at this Github repository.

3.1.2. DATASETS

The more common datasets being used to develop NIDS consist of the KDD CUP 1999, NSL-KDD, CICIDS-2017, and UNSW-NB15 (Khanan et al., 2024). These datasets are composed of various types of attacks and features observed within network traffic. Both KDD CUP 1999 and NSL-KDD were criticized back in 2015 by Moustafa (Moustafa & Slay, 2015) for not being up to date with contemporary network traffic patterns. Ahmad & Alsmadi, along with Keserwani et al. (2023), underscored the need for contemporary datasets that reflect current attack patterns to ensure the development of effective IDSs (R. Ahmad & Alsmadi, 2021). Other authors, such as Tripathy & Behera (2023), have also suggested that IDSs should be developed using larger and more diverse datasets, pushing the need for innovation in dataset choice. The UNSW-NB15 has remained as a relevant option for open-source historical network traffic data, alongside the CICIDS 2017 dataset (Khanan et al., 2024). There are other less used datasets, which are typically specialized in specific data sources or types of cyber-attacks, such as Bot-IoT (IoT-specific) or Caida 2017 (DDoS attack type) (Khanan, A., 2024). Together with previously mentioned studies, the literature seems to converge on the relevance of the UNSW-NB15 and CICIDS-2017 datasets as widely accepted benchmarks for current IDS research, offering diverse and complementary attack patterns (Khanan et al., 2024; Thakkar & Lohiya, 2023). These datasets were utilized to develop the proposed model.

3.1.3. UNSW-NB15 DATASET

The UNSW-NB15 introduced by Moustafa & Slay (2015) and was produced by monitoring communications between routers in a simulated network using the tcpdump tool. The IXIA tool

simulated the infection of one server and produced attack traffic patterns. In total, UNSW-NB15 contains a total of 31 hours of network data, across 4 separate csv files. The dataset is comprised of 49 features which have been extracted from the captured PCAP files using tools such as Argus and Bro-IDS. The features include data on the flow, content and connection as well as the time dependence of the communications. The labels present in this dataset include normal and attack patterns (Label) as well as specific attack categories amounting to 10 different classes (9 attack and 1 normal) suitable for a multiclass classification scenario (attack_cat) as well as binary classification (Moustafa & Slay, 2015).

3.1.4. CICIDS-2017 DATASET

The CICIDS-2017 dataset was introduced by Sharafaldin et al. (2018). It was generated by simulating interactions between a Victim-Network, equipped with routers and firewalls, and an Attack-Network, where diverse attack scenarios were executed using a variety of tools — described by the authors in their paper (Sharafaldin et al., 2018). The data was captured across the span of 5 days and included labels for both benign traffic and 6 of the main categories of attacks considered by the authors (Sharafaldin et al., 2018). In total, CICIDS-2017 comprises over 80 features extracted from PCAP files using the CICFlowMeter feature extractor tool. These features include metrics such as packet size and time dependency data, providing and extensive set of categories for network traffic (Sharafaldin et al., 2018).

3.1.5. DATA PREPROCESSING

Data from both datasets was initially preprocessed and potential issues were outlined by performing exploratory data analysis. Visualization techniques such as histograms, box plots and count plots were used to get a thorough understanding of features' distribution across the different target class values. Additionally, class imbalance was also identified across both datasets, an issue which will be later reflected in the evaluation metrics chosen and the additional processing steps.

During the initial development of the model, outlier treatment, using interquartile range method, was experimented with. The results seemed to negatively impact the model classification accuracy, and, therefore, this process was removed from the pipeline.

On UNSW-NB15, the categorical features which had a high cardinality were normalized to retain only 6 values with higher counts and replace the outliers with a placeholder value. This step enabled the usage of one hot encoded version of these features without producing a dimensionality issue within the dataset. All categorical features were transformed using one-hot encoding with safeguards in place to ignore unseen categories in the test set and maintain consistent feature dimensions across training and testing data.

Table 3-1 - Table Containing the classes present in each dataset

Attack Type	UNSW-NB15	CICIDS-2017
Analysis	Yes	No
Backdoors	Yes	No
Botnet ARES	No	Yes
Brute Force	Yes	Yes (e.g., FTP-Patator, SSH-Patator)
DoS/DDoS	Yes (DoS only)	Yes (DoS + DDoS)
Exploits	Yes	No
Fuzzers	Yes	No
Generic	Yes	No
Heartbleed	No (Merged under Infiltration)	Yes (within Infiltration)
Infiltration	Yes	Yes (includes Heartbleed)

Normal	Yes (BENIGN class)	Yes (BENIGN class)
PortScan	No (Grouped under Reconnaissance)	Yes
Reconnaissance	Yes	No (called PortScan)
Shellcode	Yes	No
Web Attack	No	Yes (Web Attack – Brute Force, XSS, SQLi)
Worms	Yes	No

To address the skewness of the data — filtering for features with an absolute skewness greater than one — a log transformation (log1p) was applied, ensuring non-negative values were transformed to maintain numerical stability and avoid extremely high values.

The datasets were then split into training and testing subsets, with all transformations consistently applied to both sets to prevent data leakage. Multiple normalization approaches were adopted, starting with a QuantileTransformer to map data to a normal distribution, followed by standardization using StandardScaler, and finally, normalization with MinMaxScaler. This layering of scalers helped to mitigate the effects of outliers without negatively impacting the model's performance and scale features to a common range between 1 and 0, suitable for neural networks.

Label encoding was applied to the target classes to convert it into the necessary format, and care was taken to preserve the encoder for use during model deployment and evaluation, to retrieve the original labels.

3.1.6. FEATURE SELECTION

Feature selection is an important step in the development of a classification model, as it ensures that only the most relevant features are utilized by the algorithm to make accurate predictions. This process results in a refined subset of the dataset, where the selected features are considered to hold the highest predictive power based on the method applied.

Considering what is seen in the literature, this project explored multiple feature selection strategies commonly used in the field of intrusion detection. Three techniques were chosen for evaluation: ANOVA, Random Forest, and the XGBoost embedded method.

To assess the performance and impact of each feature selection method, a systematic comparison was conducted using cross-validation on multiple feature count thresholds. For each technique, feature subsets of 30, 35, 40, 45, and 50 features were selected and evaluated, with the goal being to see which one provided the best results with the classification algorithms used in this project. To thoroughly evaluate the performance of each version of the model, using different feature selection techniques, 5-folds cross validation was applied, with each fold containing 200 epochs.

3.1.7. TREATING CLASS IMBALANCE

Class imbalance is a well-known challenge that can significantly impact the performance of classification models. In an ideal setting, training data would include a balanced number of instances for each class, allowing the algorithm to learn all target class representations equally. However, in many real-world intrusion detection datasets, certain classes are underrepresented, which is the case of both UNSW-NB15 and CICIDIS2017. This imbalance often leads to models that are biased toward the majority class, failing to detect rare but critical instances such as specific attack types. To address this issue, oversampling methods are widely adopted in literature as a strategy to balance the class distribution in the training data.

In this study, oversampling was applied after feature selection, a choice aimed at reducing the amount of noise introduced during synthetic data generation. By first filtering the dataset to

retain only the most relevant features, the synthetic samples produced through oversampling were based on a more relevant feature space. This approach aimed at amplifying solely the most relevant patterns of data.

Following the same methodology used for feature selection, 5-fold cross-validation was applied to determine the best option of resampling techniques to apply to this classification problem. Four resampling techniques were compared: ADASYN, SMOTE, SMOTEENN, and Borderline-SMOTE. All techniques were evaluated using their default parameter configurations, except for the sampling strategy, which was tailored to focus on oversampling all non-majority classes. The primary objective of this process was to increase the representation of the minority classes, thereby aligning the class weight more closely with that of the majority class and enhancing the model's ability to generalize across all categories.

3.1.8. BASE ALGORITHM USED FOR MODELLING

Neural networks trace their origins to the work of McCulloch and Pitts (1943), who first translated biological neural activity into a mathematical model, introducing the concept of artificial neurons. These foundational ideas were later expanded upon by Rosenblatt (1958), who proposed the perceptron, an artificial neuron capable of processing information based on weighted inputs. The work of Rumelhart et al. (1986) introduced the backpropagation algorithm, which enabled learning in neural networks by adjusting the weights through error minimization, paving the way for the development of the multilayer perceptron (MLP) and ultimately the concept of deep neural networks (LeCun et al., 2015). These networks would come to be identified as an expression of the universal approximation theorem (Hornik et al., 1989). The term neural network often is used to encompass algorithms with layered architectures composed by neurons, such as Convolutional Neural Networks, for example (LeCun et al., 2015).

In 1957 Kolmogorov Arnold proposed the representation theorem which stated that every multivariate continuous function can be decomposed into the superposition of simpler univariate functions (Kolmogorov, 1957). Hecht-Nielsen leveraged the Kolmogorov-Arnold network to suggest its application onto the field of neural networks (Hecht-Nielsen et al., 1987), following the developments that were being made at the time by Rumelhart et al. (Hecht-

Nielsen et al., 1987; Rumelhart et al., 1986). Since then, the idea of applying the superposition of simple functions to represent more complex functions has been explored by various authors (Köppen, 2002; Lai & Shen, 2024; Lin & Unbehauen, 1993; Montanelli & Yang, 2020). However, until now, researchers had not found a viable translation into an algorithm that could be backpropagated due to the non-smooth nature of the functions being utilized and the parameterization of the network (Z. Liu et al., 2024). In June 2024, Liu et al. introduced a novel deep learning algorithm based on the Kolmogorov-Arnold representation theorem, providing a solution to the previously identified challenges associated with KANNs by adapting the use of splines (Lai & Shen, 2024; Z. Liu et al., 2021).

Kolmogorov-Arnold Neural Networks (KANN) have since been presented as an alternative to the mainstream MLP utilizing the universal approximation theorem (Hornik et al., 1989; Z. Liu et al., 2024). In KANN most learnable linear weights are abandoned and the algorithm learns locally adjustable activation functions called splines (Liu et al., 2024). The authors highlight the strength of explainability in KANNs due to the algorithm using activation functions for approximating functions of the data, instead of weights (Liu et al., 2024).

This algorithm is hypothesized to surpass the common MLP in both accuracy and interpretability (Z. Liu et al., 2024) These characteristics of KANN make them a promising tool to be researched in the variety of fields MLP have also seen application. Therefore, the main objective of this thesis is to develop and evaluate a model for intrusion detection based on the novel Kolmogorov-Arnold Neural Network (KAN) algorithm, as previously presented. To achieve this, the implementation of KANN provided by Blealtan (2024) was used as the foundation for building the model.

There are practical implications for adopting the KANN algorithm. First, at the time of production of this thesis, there is still no deployment of a stable version of this algorithm onto any of the more popular modules used for machine learning projects. This makes it mandatory to utilize the code available on public repositories to work solely on the deployment of this algorithm onto case problem. Second, even though the authors of the most recent theoretical paper introducing KANNs provide great advice on how to scale KANNs in comparison to neural networks, KANNs retain the uncertainties present in neural networks in general when it comes to what architecture best solves a problem. Additionally, KANNs can be scaled not only in terms of depth and width, as in traditional neural networks, but also through the concept known as

fine graining. Fine-graining refers to the increased flexibility in the model by adjusting the complexity of the spline-based activation functions used in each node. By fine-tuning the number of spline segments, users control how precisely these functions can adapt to the input data during training, offering an extra layer of architectural granularity (Z. Liu et al., 2024). This in turn translates to an added layer of complexity to the way in which this algorithm is tested, requiring further testing than that of a common neural network. The architecture of the final model was fine-tuned iteratively through cross-validation, adjusting both the width and depth of the network, as well as the granularity of each node.

3.1.9. EVALUATION METRICS

The importance of selecting appropriate performance metrics is underscored by Maseer et al. (2023). These authors highlight the frequency of evaluation metrics choice being different for different classification scenarios. It is observed that for the binary classification scenario the precision metric is much more prevalent, with accuracy being the second most recurrent. The inverse is seen for multi-class classification where the authors observed accuracy to be more prevalent out of all the metrics, with precision being the second most recurrent (Maseer et al., 2023). Tripathy and Behera (2023) further emphasized the importance of accuracy as a key evaluation metric in IDS. Additionally, the authors imply the need to complement accuracy with additional metrics, such as precision, recall, and F1-score, to gain a more comprehensive understanding of model performance (Tripathy & Behera, 2023). Together, these reviews underscore the critical role of performance metrics, especially accuracy and precision, in developing and evaluating robust and reliable IDS models.

In this section, I present the evaluation metrics that will be used during the evaluation part of this thesis. These metrics include accuracy, precision, recall, F1-Score and Macro F1-Score a variety of metrics appropriate to evaluate the classification performance in a data imbalance scenario. Each metric is derived from the following components:

i. True Positive (TP): The data instances correctly predicted as an Attack
ii. False Negative (FN): The data instances wrongly predicted as Normal instances.
iii. False Positive (FP): The data instances wrongly classified as an Attack.

iv. True Negative (TN): The instances correctly classified as Normal instances.

Formulas

- **Accuracy**

  Accuracy measures the proportion of correctly classified instances out of the total number of instances (Sokolova & Lapalme, 2009)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **F1-Score**

  F1-score is the harmonic mean between precision and recall (Sokolova & Lapalme, 2009)

$$F1 - Score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

- **Precision**

  Precision evaluates the proportion of correctly identified attacks among all instances predicted as attacks (Sokolova & Lapalme, 2009)

$$Precision = \frac{TP}{TP + FP}$$

- **Recall**

  Recall measures the proportion of actual attacks correctly identified by the model (Sokolova & Lapalme, 2009)

$$Recall = \frac{TP}{TP + FN}$$

- **Macro F1-Score**

  Macro F1-score calculates the F1-score independently for each class and then takes the

unweighted average, treating all classes equally regardless of their size (Sokolova & Lapalme, 2009)

$$Macro\ F1 - Score\ = \frac{(F1^1 + F1^2 + \ldots + F1_n)}{n}$$

4. RESULTS AND DISCUSSION

To proposed methodology allowed us to reach various results across, producing different versions of the final model before reaching a final solution. As mentioned in the above chapters, cross-validation was used to evaluate the different versions on a specified set of metrics which best fit the characteristics of a multi-class classification problem, with class imbalance.

The feature selection technique which provided the highest score across the chosen evaluation metrics was XGBOOST, for both models, as can be seen in table 4.1. Additionally, it can be noted that ANOVA will often provide worse results when compared to the other tree-based methods. The resampling technique which provided the highest results in terms of evaluation metrics for both models was SMOTEENN and thus was chosen to be used for the final two models. Additionally, altering the parameters for SMOTEENN did not affect the results obtained for CICIDS2017, however, it did prove beneficial for UNSW-NB15.

Model Version	Accuracy	Weighted F1-Score	Macro F1-Score
45 Features RF	99.30	99.29	91.45
40 Features XGBOOST	99.29	99.27	89.83
35 Features RF	99.26	99.25	90.70
35 Features XGBOOST	99.25	99.24	92.60
50 Features XGBOOST	99.25	99.24	90.62
45 Features XGBOOST	99.25	99.24	89.71
50 Features RF	99.24	99.23	90.47
40 Features RF	99.21	99.20	88.43
30 Features XGBOOST	98.36	98.35	89.60
50 Features ANOVA	98.30	98.29	86.73
45 Features ANOVA	98.30	98.28	85.61
30 Features RF	98.25	98.24	86.32
40 Features ANOVA	98.20	98.18	89.31
35 Features ANOVA	98.16	98.14	87.60
30 Features ANOVA	95.90	95.88	84.46

Figure 4-1 Feature Selection Technique results for CICIDS-2017

The resulting model was finetuned through iteration and consists of a neural network with three hidden layers, containing 20, 10, and 5 neurons, respectively. Each neuron was configured with a grid size of 6 and a spline order of 3. To accommodate the specific requirements of each dataset, the input and output layers were adjusted accordingly.

Model Version	Accuracy	Weighted F1-Score	Macro F1-Score
45 Features XGBOOST	81.78	79.48	51.28
50 Features XGBOOST	81.89	79.36	50.93
50 Features RF	81.89	79.36	50.93
45 Features RF	81.55	79.35	51.21
50 Features ANOVA	81.75	79.34	51.07
40 Features XGBOOST	81.80	79.26	51.24
30 Features XGBOOST	81.63	79.09	50.8
35 Features XGBOOST	81.70	79.07	50.72
40 Features RF	81.63	79.07	50.57
45 Features XGBOOST	81.03	78.77	49.15
35 Features RF	81.19	78.55	49.16
30 Features RF	81.21	78.50	48.53
40 Features ANOVA	81.07	78.48	48.08
35 Features ANOVA	80.04	77.41	47.17
30 Features ANOVA	78.85	76.10	43.72

Figure 4-2 Feature Selection Technique results for UNSW-NB15

It can be concluded that the models developed in this thesis, especially CICIDS2017, perform competitively compared to other models presented in literature, particularly in multiclass classification scenarios.

Model Version	Dataset	Avg val Acc	Avg val Weighted F1-Score	Avg val Macro F1-Score
35 Features ADASYN	CICIDS2017	99.54	99.54	99.53
35 Features SMOTEEN (n=3)	CICIDS2017	99.63	99.63	99.63
35 Features Borderline SMOTE	CICIDS2017	99.59	99.59	99.59
35 Features SMOTE	CICIDS2017	99.61	99.61	99.61
45 Features ADASYN	UNSW-NB15	64.80	65.25	65.08
45 Features SMOTEEN (n=3)	UNSW-NB15	77.96	78.49	78.61
45 Features Bordeline SMOTE	UNSW-NB15	69.95	70.55	70.55
45 Features SMOTE	UNSW-NB15	73.24	73.87	73.87
35 Features SMOTEEN (n=5)	**CICIDS2017**	**99.63**	**99.63**	**99.63**
45 Features SMOTEEN (n=5)	**UNSW-NB15**	**78.57**	**78.71**	**79.19**

Figure 4-3 Final Results for each XGBoost KANN after resampling

It can be observed that a shift in the metrics is seen before and after applying the resampling techniques. The results show that, for the case of UNSW-NB15, the accuracy score is lowered

marginally while the average macro f1-score jumps significantly. This can be seen as the model being able to more equally classify different classes that are present in the dataset. Therefore, it may be concluded that there is a major issue with the UNSW-NB15 dataset which sees a partial resolution from applying resampling techniques. In general, the application of resampling techniques proved beneficial to the model across both datasets, with a higher impact for UNSW-NB15.

Regarding UNSW-NB15, the best model obtained an accuracy of 78.57%, a weighted F1-score of 78.71 and a macro F1-score of 79.19. The discrepancy between the overall results seen between F1-score and weighted F1-score reveal the major problematic present in these datasets, which is the imbalance issue. However, considering the diversity in labels, the results obtained still display a significant capacity in the developed model's prediction power.

Regarding CICIDS2017, the best model obtained an accuracy of 99.63%, a weighted F1-score of 99.63% and macro F1-score of 99.63%. The overall metrics obtained for CICIDS2017 using a KANN model remain competitive, even across different techniques for oversampling and feature selection. In comparison to UNSW-NB15, the process of oversampling does not prove to be crucial to improve significantly the model's results, however still displaying a marginal increase in all metrics. A strong point for this version of our KANN model is the increase in macro F1-score, boosting the model to be competitive in predicting the variety of classes available in the dataset.

However, it is valuable to contextualize the performance of these models within the literature lifted in previous chapters. A key strength of the developed models lies in their ability to maintain competitive performance in multiclass classification scenarios and that will be highly considered for the comparisons made moving forward.

The dataset complexity serves as a point for comparison. Most high-performing models in literature either rely on less contemporary and robust datasets (NSL-KDD, KDD99) or narrowly scoped datasets (e.g., MQTT, N-BaIoT) whereas this study targets two of the most widely recognized and comprehensive benchmarks: UNSW-NB15 and CICIDS2017 (Khanan, A., 2024). Models such as the ones proposed by Nguyen et al. (2022) and Sharma et al. (2023) demonstrate strong results, they operate on binary, reduced-class settings or lack transparency in multiclass metric report, limiting comparability. In contrast, my KANN model maintains consistent and transparent reporting across all metrics, with CICIDS2017 reaching a performance of 99.63% across accuracy, weighted F1, and macro F1-score, results more uncommonly observed in comparable models without relying on ensemble architectures or complex attention mechanisms.

The treatment of class imbalance further distinguishes the proposed model. Unlike Yang et al. (2019), who opt for CVAE-based augmentation, or Sharma et al. (2023), who employ GANs, the present study focuses on simpler method like SMOTEEN, presenting similarly competitive results with less training complexity. The study critically evaluates the impact of resampling by comparing performance before and after oversampling, this is something that is often omitted or underexplored in literature. The observed shift in UNSW-NB15, where accuracy slightly decreases but macro F1-score significantly improves, reveals and improvement to the model's real-world applicability.

In terms of architectural complexity, this work illustrates that strong performance while resorting to a relatively simple architecture. My model, with its compact topology (three hidden layers with 20, 10, and 5 neurons respectively), achieves results comparable with deeper or ensemble-based models. Whereas models like El-Rady et al. (2023) and Le et al. (2022) employ CNNs with hundreds of thousands of parameters, often paired with GANs for oversampling, this study leverages KAN's and proposes a much simpler approach. This argument could be further developed with a study that focuses more on the comparison of learnable parameters present in each model.

Feature selection and resampling were given a significant focus in this thesis. While some works adopt basic feature filtering or rely on default feature sets (e.g., Mahmood & Al Dabagh, 2023), this thesis evaluates three distinct selection techniques (XGBoost, ANOVA, Random Forest) using multiple feature thresholds while resorting to cross-validation. The consistent superiority of XGBoost aligns with findings by Sajid et al. (2024) and Liu et al. (2021) and reinforces the importance of leveraging embedded tree-based selectors in high-dimensional, and scenarios in which outliers are abundant, like IDS.

5. CONCLUSION

This thesis proposed and developed a novel Network Intrusion Detection System (NIDS) based on the Kolmogorov-Arnold Neural Network (KANN) algorithm, applied to two benchmark datasets: UNSW-NB15 and CICIDS2017. The study addressed a key gap in the current literature by investigating the viability of KANNs—recently introduced deep learning models based on the Kolmogorov-Arnold representation theorem—as a transparent and competitive alternative to more traditional neural networks, particularly in the context of intrusion detection.

A thorough methodology was established, guided by the CRISP-DM framework, which included data preprocessing, three feature selection strategies (XGBoost, Random Forest, and ANOVA), and four oversampling techniques (ADASYN, SMOTE, Borderline SMOTE, and SMOTEENN). The consistent performance improvement observed when using XGBoost for feature selection and SMOTEENN for resampling demonstrated the importance of preprocessing in IDS development. Cross-validation and multiple evaluation metrics—accuracy, F1-score, and macro F1-score—were applied to ensure rigorous assessment, particularly given the class imbalance present in both datasets.

The findings show that the developed KANN-based models perform competitively when compared to existing models in the literature, especially in multi-class classification scenarios. For CICIDS2017, the model achieved 99.63% across all primary evaluation metrics, rivaling more complex architectures such as ensembles and attention-based systems, while maintaining a much simpler structure. For UNSW-NB15, despite greater challenges due to class imbalance, the final model reached a macro F1-score of 79.19%, highlighting its capacity to generalize across a diverse set of attack classes. The shift in performance—where accuracy slightly decreased but macro F1-score improved—underscores the positive impact of resampling in addressing imbalance and enhancing real-world applicability.

In terms of originality, this study represents the first structured evaluation of KANNs applied to these widely used IDS datasets. By maintaining a focus on simplicity and methodological clarity, this thesis contributes to the broader discourse on how to design more compact and resource-efficient deep learning-based IDSs—an increasingly relevant consideration in IoT and fog computing environments.

Several limitations must be acknowledged. First, despite the theoretical promise of KANNs in delivering interpretable outputs, the practical exploration of their explainability was limited by the current lack of available tools and APIs to visualize or interpret the learned spline functions, leaving XAI outside the scope for this thesis. Second, this thesis did not include a formal comparison of computational complexity or training efficiency against other models, which limits the conclusions that can be drawn about KANNs' suitability for resource constrained environments.

The implications of this research are both practical and theoretical. Practically, the study demonstrates that KANNs can achieve strong predictive performance while retaining compact architecture and using fewer parameters than most deep learning counterparts. Theoretically, the

results suggest that spline-based function learning may be a promising direction for future IDS algorithms, particularly in contexts where interpretability is valued.

Future research should expand in the following directions:

- Development of explainability modules to visualize and analyze KANN spline functions, enabling their use in XAI applications.
- Formal benchmarking of KANN computational efficiency
- Exploration of other detection techniques, such as hybrid architectures, using KANN as a base model.

In summary, this work contributes a new perspective to IDS research by introducing and validating a novel deep learning algorithm thorough and replicable way. The demonstrated performance, combined with architectural simplicity and methodological transparency, marks this thesis as a valuable foundation for future studies on explainable and efficient network intrusion detection systems.

6. BIBLIOGRAPHICAL REFERENCES

Abdul Lateef, A. A., Faraj Al-Janabi, S. T., & Al-Khateeb, B. (2020). Hybrid Intrusion Detection System Based on Deep Learning. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 1–5. https://doi.org/10.1109/ICDABI51230.2020.9325669

Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. *Electronics*, *11*(2), 198. https://doi.org/10.3390/electronics11020198

Ahmad, R., & Alsmadi, I. (2021). Machine learning approaches to IoT security: A systematic literature review. *Internet of Things*, *14*, 100365. https://doi.org/10.1016/j.iot.2021.100365

Ahmad, R., Alsmadi, I., Alhamdani, W., & Tawalbeh, L. (2023). Zero-day attack detection: A systematic literature review. *Artificial Intelligence Review*, *56*(10), 10733–10811. https://doi.org/10.1007/s10462-023-10437-z

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

Al, S., & Dener, M. (2021). STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Computers & Security*, *110*, 102435. https://doi.org/10.1016/j.cose.2021.102435

Albulayhi, K., Smadi, A. A., Sheldon, F. T., & Abercrombie, R. K. (2021). IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors*, *21*(19), 6432. https://doi.org/10.3390/s21196432

Alsaleh, S. S., El Bachir Menai, M., & Al-Ahmadi, S. (2024). Federated Learning-Based Model to Lightweight IDSs for Heterogeneous IoT Networks: State-of-the-Art, Challenges, and Future Directions. *IEEE Access*, *12*, 134256–134272. https://doi.org/10.1109/ACCESS.2024.3460468

Amouri, A., Rahhal, M. M. A., Bazi, Y., Butun, I., & Mahgoub, I. (2024). *Enhancing Intrusion Detection in IoT Environments: An Advanced Ensemble Approach Using Kolmogorov-Arnold Networks* (arXiv:2408.15886). arXiv. http://arxiv.org/abs/2408.15886

Arik, S. Ö., & Pfister, T. (2021). TabNet: Attentive Interpretable Tabular Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(8), Artigo 8. https://doi.org/10.1609/aaai.v35i8.16826

Arisdakessian, S., Wahab, O. A., Mourad, A., Otrok, H., & Guizani, M. (2023). A Survey on IoT Intrusion Detection: Federated Learning, Game Theory, Social Psychology, and Explainable AI as Future Directions. *IEEE Internet of Things Journal*, *10*(5), 4059–4092. https://doi.org/10.1109/JIOT.2022.3203249

Arivazhagan., C., & Natarajan., V. (2020). A Survey on Fog computing paradigms, Challenges and Opportunities in IoT. *2020 International Conference on Communication and Signal Processing (ICCSP)*, 0385–0389. https://doi.org/10.1109/ICCSP48568.2020.9182229

Assy, A. T., Mostafa, Y., El-Khaleq, A. A., & Mashaly, M. (2023). *Anomaly-Based Intrusion Detection System using One-Dimensional Convolutional Neural Network*. *220*, 78–85. Scopus. https://doi.org/10.1016/j.procs.2023.03.013

Axelsson, S. (2000). *Intrusion Detection Systems: A Survey and Taxonomy*.

Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog Computing: A Platform for Internet of Things and Analytics. Em N. Bessis & C. Dobre (Eds.), *Big Data and Internet of Things: A Roadmap for Smart Environments* (Vol. 546, pp. 169–186). Springer International Publishing. https://doi.org/10.1007/978-3-319-05029-4_7

Bostani, H., & Sheikhan, M. (2017). Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Computer Communications*, *98*, 52–71. https://doi.org/10.1016/j.comcom.2016.12.001

Denning, D. E. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, *SE-13*(2), 222–232. https://doi.org/10.1109/TSE.1987.232894

Erza Aminanto, M., Rifqi Purbomukti, I., Chandra, H., & Kim, K. (2022). Two-Dimensional Projection-Based Wireless Intrusion Classification Using Lightweight EfficientNet. *Computers, Materials & Continua*, *72*(3), 5301–5314. https://doi.org/10.32604/cmc.2022.026749

Hecht-Nielsen, R., Drive, O., & Diego, S. (1987). *Kolmogorov's Mapping Neural Network Existence Theorem*.

Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., Sharma, B., & Chowdhury, S. (2023). Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. *Sensors*, *23*(2). Scopus. https://doi.org/10.3390/s23020890

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Houda, Z. A. E., Brik, B., & Khoukhi, L. (2022). "Why Should I Trust Your IDS?": An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks. *IEEE*

Open Journal of the Communications Society, *3*, 1164–1176. https://doi.org/10.1109/OJCOMS.2022.3188750

Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: Architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*, *98*, 27–42. https://doi.org/10.1016/j.jnca.2017.09.002

Jamalipour, A., & Murali, S. (2022). A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey. *IEEE Internet of Things Journal*, *9*(12), 9444–9466. https://doi.org/10.1109/JIOT.2021.3126811

Kamaldeep, Dutta, M., & Granjal, J. (2020). Towards a Secure Internet of Things: A Comprehensive Study of Second Line Defense Mechanisms. *IEEE Access*, *8*, 127272–127312. https://doi.org/10.1109/ACCESS.2020.3005643

Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, *199*, 113–125. Scopus. https://doi.org/10.1016/j.comcom.2022.12.010

Keserwani, P. K., Govil, M. C., & Pilli, E. S. (2023). An effective NIDS framework based on a comprehensive survey of feature optimization and classification techniques. *Neural Computing and Applications*, *35*(7), 4993–5013. https://doi.org/10.1007/s00521-021-06093-5

Khanan, A., Abdelgadir Mohamed, Y., Mohamed, A. H. H. M., & Bashir, M. (2024). From Bytes to Insights: A Systematic Literature Review on Unraveling IDS Datasets for Enhanced Cybersecurity Understanding. *IEEE Access*, *12*, 59289–59317. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3392338

Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. Em V. Arnol'd, V.

Boltjanskiĭ, N. Efimov, G. Èskin, A. Kolmogorov, D. Koteljanskiĭ, N. Krasovskiĭ, D. Men'šov, I. Portnov, S. Ryškov, Ju. Šaškin, G. Šilov, S. Stečkin, S. Teljakovskiĭ, N. Trebukova, & N. Vilenkin, *American Mathematical Society Translations: Series 2* (Vol. 28, pp. 55–59). American Mathematical Society. https://doi.org/10.1090/trans2/028/04

Köppen, M. (2002). On the Training of a Kolmogorov Network. Em J. R. Dorronsoro (Ed.), *Artificial Neural Networks—ICANN 2002* (Vol. 2415, pp. 474–479). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46084-5_77

Kumar, M., & Singh, A. (2020). *Distributed Intrusion Detection System using Blockchain and Cloud Computing Infrastructure*. 248–252. https://doi.org/10.1109/ICOEI48184.2020.9142954

Lai, M.-J., & Shen, Z. (2024). *The Kolmogorov Superposition Theorem can Break the Curse of Dimensionality When Approximating High Dimensional Functions* (arXiv:2112.09963). arXiv. https://doi.org/10.48550/arXiv.2112.09963

Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S., Hosseinzadeh, M., & Rahmani, A. M. (2021). Deep Learning-Based Intrusion Detection Systems: A Systematic Review. *IEEE Access*, *9*, 101574–101599. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3097247

Le, A., Loo, J., Chai, K., & Aiash, M. (2016). A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology. *Information*, *7*(2), 25. https://doi.org/10.3390/info7020025

Le, K.-H., Nguyen, M.-H., Tran, T.-D., & Tran, N.-D. (2022). IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT. *Electronics*, *11*(4), 524. https://doi.org/10.3390/electronics11040524

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*, *7*, 8176–8186. https://doi.org/10.1016/j.egyr.2021.08.126

Lin, J.-N., & Unbehauen, R. (1993). On the Realization of a Kolmogorov Network. *Neural Computation*, *5*(1), 18–20. https://doi.org/10.1162/neco.1993.5.1.18

Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*, *9*(20), 4396. https://doi.org/10.3390/app9204396

Liu, Z., Thapa, N., Shaver, A., Roy, K., Siddula, M., Yuan, X., & Yu, A. (2021). Using Embedded Feature Selection and CNN for Classification on CCD-INID-V1—A New IoT Dataset. *Sensors*, *21*(14), Artigo 14. https://doi.org/10.3390/s21144834

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). *KAN: Kolmogorov-Arnold Networks* (arXiv:2404.19756). arXiv. http://arxiv.org/abs/2404.19756

Mahmood, M. S., & Al Dabagh, N. B. (2023). Improving IoT Security using Lightweight Based Deep Learning Protection Model. *Tikrit Journal of Engineering Sciences*, *30*(1), 119–129. https://doi.org/10.25130/tjes.30.1.12

Maithem, M., & Al-Sultany, G. A. (2021). *Network intrusion detection system using deep neural networks*. *1804*(1). Scopus. https://doi.org/10.1088/1742-6596/1804/1/012138

Maseer, Z. K., Yusof, R., Al-Bander, B., Saif, A., & Kadhim, Q. K. (2023). *Meta-Analysis and Systematic Review for Anomaly Network Intrusion Detection Systems: Detection Methods, Dataset, Validation Methodology, and Challenges* (arXiv:2308.02805). arXiv. https://doi.org/10.48550/arXiv.2308.02805

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, *17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

Momand, A., Jan, S. U., & Ramzan, N. (2023). A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning: Deep Learning, Datasets, and Attack Taxonomy. *Journal of Sensors*, *2023*, 1–18. https://doi.org/10.1155/2023/6048087

Montanelli, H., & Yang, H. (2020). *Error bounds for deep ReLU networks using the Kolmogorov—Arnold superposition theorem* (arXiv:1906.11945). arXiv. https://doi.org/10.48550/arXiv.1906.11945

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

Mukhaini, G. A., Anbar, M., Manickam, S., Al-Amiedy, T. A., & Momani, A. A. (2024). A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks. *Journal of King Saud University - Computer and Information Sciences*, *36*(1), 101866. https://doi.org/10.1016/j.jksuci.2023.101866

Muneer, S., Farooq, U., Athar, A., Ahsan Raza, M., Ghazal, T. M., & Sakib, S. (2024). A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. *Journal of Engineering*, *2024*, 1–16. https://doi.org/10.1155/2024/3909173

Murugesh, C., & Murugan, S. (2023). Moth Search Optimizer with Deep Learning Enabled Intrusion Detection System in Wireless Sensor Networks. *SSRG International Journal of Electrical and*

Electronics Engineering, *10*(4), 77–90. Scopus. https://doi.org/10.14445/23488379/IJEEE-V10I4P108

Najafli, S., Toroghi Haghighat, A., & Karasfi, B. (2024). Taxonomy of deep learning-based intrusion detection system approaches in fog computing: A systematic review. *Knowledge and Information Systems*, *66*(11), 6527–6560. https://doi.org/10.1007/s10115-024-02162-y

Nguyen, T.-N., Dang, K.-M., Tran, A.-D., & Le, K.-H. (2022). Towards an Attention-Based Threat Detection System for IoT Networks. Em T. K. Dang, J. Küng, & T. M. Chung (Eds.), *Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications* (Vol. 1688, pp. 301–315). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-8069-5_20

Otoum, Y., & Nayak, A. (2021). AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. *Journal of Network and Systems Management*, *29*(3), 23. https://doi.org/10.1007/s10922-021-09589-6

Puliafito, C., Mingozzi, E., Longo, F., Puliafito, A., & Rana, O. (2019). Fog Computing for the Internet of Things: A Survey. *ACM Transactions on Internet Technology*, *19*(2), 1–41. https://doi.org/10.1145/3301443

Pundir, S., Wazid, M., Singh, D. P., Das, A. K., Rodrigues, J. J. P. C., & Park, Y. (2020). Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges. *IEEE Access*, *8*, 3343–3363. https://doi.org/10.1109/ACCESS.2019.2962829

Radouan Ait Mouha, R. A. (2021). Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, *09*(02), 77–101. https://doi.org/10.4236/jdaip.2021.92006

Rashid Abdulqadir, H., R. M. Zeebaree, S., M. Shukur, H., Mohammed Sadeeq, M., Wasfi Salim, B., Abid Salih, A., & Fattah Kak, S. (2021). A Study of Moving from Cloud Computing to Fog

Computing. *Qubahan Academic Journal*, *1*(2), 60–70. https://doi.org/10.48161/qaj.v1n2a49

Rashid, M. M., Kamruzzaman, J., Hassan, M. M., Imam, T., & Gordon, S. (2020). Cyberattacks Detection in IoT-Based Smart City Applications Using Machine Learning Techniques. *International Journal of Environmental Research and Public Health*, *17*(24), 9347. https://doi.org/10.3390/ijerph17249347

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. Scopus. https://doi.org/10.1038/323533a0

Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing*, *13*(1), 123. https://doi.org/10.1186/s13677-024-00685-x

Samann, F. E. F., Abdulazeez, A. M., & Askar, S. (2021). Fog Computing Based on Machine Learning: A Review. *International Journal of Interactive Mobile Technologies (iJIM)*, *15*(12), 21. https://doi.org/10.3991/ijim.v15i12.21313

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science*, *171*, 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization: *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 108–116. https://doi.org/10.5220/0006639801080116

Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. *Computers and Electrical Engineering*, *107*. Scopus. https://doi.org/10.1016/j.compeleceng.2023.108626

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Thakkar, A., & Lohiya, R. (2023). Attack Classification of Imbalanced Intrusion Data for IoT Network Using Ensemble-Learning-Based Deep Neural Network. *IEEE Internet of Things Journal*, *10*(13), 11888–11895. Scopus. https://doi.org/10.1109/JIOT.2023.3244810

Tripathy, S. S., & Behera, B. (2023). *PERFORMANCE EVALUATION OF MACHINE LEARNING ALGORITHMS FOR INTRUSION DETECTION SYSTEM* (2023/1546). Cryptology ePrint Archive. https://eprint.iacr.org/2023/1546

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network. *Sensors*, *19*(11), 2528. https://doi.org/10.3390/s19112528

Zhao, R., Gui, G., Xue, Z., Yin, J., Ohtsuki, T., Adebisi, B., & Gacanin, H. (2022). A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things. *IEEE Internet of Things Journal*, *9*(12), 9960–9972. https://doi.org/10.1109/JIOT.2021.3119055