ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A SUPPORT VECTOR MACHINE-BASED APPROACH FOR SOUTHBOUND COMMUNICATION DETECTION IN SDN USING OPENFLOW

M.Sc. THESIS

Ali Gökhan AVRAN

Department of Computer Engineering

Computer Engineering Programme

FEBRUARY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A SUPPORT VECTOR MACHINE-BASED APPROACH FOR SOUTHBOUND COMMUNICATION DETECTION IN SDN USING OPENFLOW

M.Sc. THESIS

Ali Gökhan Avran
(504201507)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Asst. Prof. Gökhan Seçinti

FEBRUARY 2024

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

OPENFLOW KULLANARAK SDN'DE GÜNEY YÖNLÜ İLETİŞİM TESPİTİ İÇİN DESTEK VEKTÖR MAKİNESİ TABANLI BİR YAKLAŞIM

YÜKSEK LİSANS TEZİ

Ali Gökhan Avran
(504201507)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Dr.Öğr.Üyesi Gökhan Seçinti

ŞUBAT 2024

Ali Gökhan Avran, a M.Sc. student of İTU Graduate School student ID 504201507, successfully defended the thesis/dissertation entitled "A SUPPORT VECTOR MACHINE-BASED APPROACH FOR SOUTHBOUND COMMUNICATION DETECTİON IN SDN USING OPENFLOW", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Gökhan Seçinti** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Şule Gündüz Öğüdücü** Istanbul Technical University	.....
	Asst. Prof. Elif BOZKAYA National Defence University	.....

Date of Submission : 05 January 2024
Date of Defense : 16 February 2024

To my parents, For all their support.

FOREWORD

First and foremost, I extend my heartfelt appreciation to my supervisor, Asst. Prof. Gökhan Seçinti for her unwavering support and invaluable guidance throughout the completion of my M.Sc. thesis.

In conclusion, I wish to convey my deep gratitude to my family for their enduring support throughout my entire life.

January 2024

Ali Gökhan Avran
(Computer Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

FP **:** False Positive
FN **:** False Negative
IDS **:** Intrusion Detection Systems
NIC **:** Network Interface Card
RBF **:** Radial Basis Function
SDN **:** Software-Defined Networking
SVM **:** Support Vector Machine
TP **:** True Positive
TN **:** True Negative

SYMBOLS

M_x **:** OpenFlow Message Category Numbers

LIST OF TABLES

Page

LIST OF FIGURES

Page

A SUPPORT VECTOR MACHINE-BASED APPROACH FOR SOUTHBOUND COMMUNICATION DETECTİON IN SDN USING OPENFLOW

SUMMARY

Software-Defined Networking represents a transformative shift in the way network management and operations are conducted. Traditionally, network devices like routers and switches have both control (the intelligence to make decisions) and data (the actual forwarding of network packets) planes integrated within them. This conventional architecture often leads to complexities in network configuration and limits flexibility.

SDN addresses these challenges by decoupling the control plane from the data plane. In an SDN architecture, the control plane is centralized in an SDN controller, a software-based entity that oversees the behavior of the network. This centralization allows for more streamlined and flexible management of the network, as changes can be implemented quickly and globally through software updates rather than hardware reconfiguration.

The southbound interface in SDN is the key that enables this communication between the SDN controller and the network devices (like routers and switches) that make up the data plane. Protocols such as OpenFlow are commonly used in this interface to facilitate the transfer of information and instructions. Through the southbound interface, the SDN controller can dynamically adjust network routes, manage traffic, and implement security policies directly to the network devices.

The importance of the southbound interface, and SDN, lies in the unprecedented level of control and flexibility it provides in network management. It allows for more efficient use of network resources, quicker adaptation to changing network conditions, enhanced security through centralized control, and easier implementation of new services or changes. This makes SDN and its components like the southbound interface pivotal in modern networking, particularly in environments that demand high scalability, agility, and security.

The southbound interface is a crucial component in SDN, serving as the vital communication link between the SDN controller and the underlying network infrastructure. In the context of SDN, which is an innovative approach to network management, this interface is essential for enabling dynamic and efficient network configuration and operation. This includes a range of devices like switches, routers, and other essential network components. The primary function of this interface is to facilitate the SDN controller in requisitioning network resources and in orchestrating the behavior of the network according to specific needs and requirements.

This interface's significance in SDN architecture cannot be overstated. It is instrumental in translating the high-level, abstract policies and rules set by the SDN controller into tangible, executable commands that the network devices can understand and act upon. This translation and communication capability is crucial for realizing the

full potential of SDN, which is centered around enhanced network management, increased agility, and improved efficiency in network resource utilization.

However, the incorporation of the southbound interface in SDN architectures is not without its challenges and risks. One of the primary concerns revolves around security. The interface, being a critical communication channel, could potentially become a target for malicious activities. Unauthorized access or manipulation of this interface could lead to severe network disruptions, data breaches, and compromise of network integrity.

Another concern is related to compatibility and standardization. The diverse nature of network devices, each possibly having different communication protocols and interfaces, poses a challenge in ensuring seamless and efficient communication through the southbound interface. This necessitates the development and adoption of standardized protocols and interfaces, which can guarantee consistent and reliable communication across various network components.

Moreover, the robustness and reliability of the southbound interface are paramount. Any failure or inefficiency in this interface could lead to significant network performance issues. It is essential to design and implement this interface with high resilience and fault tolerance to ensure uninterrupted network operations.

This study presents a novel framework for packet classification in SDN using an OpenFlow Protocol and a Support Vector Machine. The central focus of this framework is to enhance the security and efficiency of southbound communication in SDN environments, a critical component in the management and operation of these networks.

The core innovation of this framework lies in its integration of SVM into the OpenFlow communication process. SVM is a well-regarded machine learning model known for its effectiveness in pattern recognition and classification tasks. In the context of SDN, the SVM is employed to analyze and classify network packets, a task essential for maintaining network security and performance.

The unique aspect of this approach is the introduction of an adjusted-weight level methodology. This novel technique involves fine-tuning the SVM model to recognize and categorize complex patterns in southbound SDN communication data more accurately. By adjusting the weight levels of different parameters in the OpenFlow protocol, the SVM model can more effectively differentiate between various types of network packets. This is particularly important in identifying and mitigating potential security threats that may be present in network traffic.

An extensive empirical analysis was conducted to validate the effectiveness of the proposed framework. The results of this analysis demonstrate a significant improvement in the accuracy of packet classification in southbound SDN communication. Specifically, the framework showed superior performance in identifying and categorizing different types of packets within the SDN environment, highlighting its potential to enhance both the security and efficiency of SDN operations.

This paper introduces a cutting-edge approach to packet classification in SDN, utilizing the strengths of SVM in conjunction with an innovative adjusted-weight level methodology. The framework promises to be a valuable tool in enhancing the security and performance of SDN environments, offering a sophisticated solution to the challenges of managing southbound communication in these networks.

The solution presented in this research has shown exceptional results in terms of accuracy and reliability, as evidenced by the impressive detection rate of 98.5%, as quantified by the classification model's score. This high level of accuracy is especially significant considering the intricate nature of network traffic in SDN environments. The ability to detect and classify network packets with such precision is a crucial advancement in the field of network security and management.

One of the most notable achievements of this framework is its remarkably low rate of false alarms. This aspect is particularly important in network management, as false alarms can lead to unnecessary disruptions and can diminish the overall efficiency of network operations. The minimization of these false positives indicates a high level of sophistication in the framework’s design and implementation, making it a reliable tool for network administrators.

The framework, centered around the OpenFlow packet classification, not only excels in its current application but also lays a strong foundation for future research. It provides a versatile and effective platform for implementing more advanced security mechanisms in SDN environments. The adaptability and effectiveness of this framework make it an ideal candidate for further exploration and development, particularly in addressing the evolving security challenges in SDN contexts.

Moreover, the framework's potential in mitigating prevalent security risks in SDN environments is highly promising. With the increasing complexity and sophistication of cyber threats, the need for robust and adaptive security solutions in SDN is more critical than ever.

Consequently, this framework, with its high detection accuracy and low false alarm rate, offers a significant step forward in meeting these security needs. Its success opens new avenues for research and development in SDN security, potentially leading to more secure and efficient network environments.

OPENFLOW KULLANARAK SDN'DE GÜNEY YÖNLÜ İLETİŞİM TESPİTİ İÇİN DESTEK VEKTÖR MAKİNESİ TABANLI BİR YAKLAŞIM

ÖZET

SDN, ağ yönetimi ve işletiminin yürütülme şeklinde dönüştürücü bir değişiklik temsil etmektedir. Geleneksel olarak, yönlendiriciler ve anahtarlar gibi ağ cihazları, hem kontrol (karar verme zekası) hem de veri (ağ paketlerinin gerçek iletimi) katmanlarını içlerinde entegre ederler. Bu geleneksel mimari, sıklıkla ağ yapılandırmasında karmaşıklıklara yol açar ve esnekliği sınırlar.

SDN, bu zorluklara, kontrol katmanını veri katmanından ayırarak yanıt verir. Bir SDN mimarisinde, kontrol katmanı bir SDN denetleyicisinde merkezileştirilmiştir. Bu, bir yazılım tabanlı varlık olup, ağın davranışını denetler. Bu merkezileştirme, değişikliklerin yazılım güncellemeleri aracılığıyla hızla ve küresel olarak uygulanabilmesi sayesinde, ağın daha akıcı ve esnek bir şekilde yönetilmesini sağlar.

SDN'deki güney yönlü iletişim, SDN denetleyicisi ile veri katmanını oluşturan ağ cihazları (örneğin yönlendiriciler ve anahtarlar) arasındaki bu iletişimi mümkün kılan anahtardır. Bu arabirimde, bilgi ve talimatların aktarımını kolaylaştırmak için genellikle OpenFlow gibi protokoller kullanılır. Güney arabirimi aracılığıyla, SDN denetleyicisi dinamik olarak ağ yollarını ayarlayabilir, trafiği yönetebilir ve ağ cihazlarına doğrudan güvenlik politikaları uygulayabilir.

Güney yönlü iletişim ve SDN'nin önemi, ağ yönetiminde sağladığı eşsiz kontrol ve esneklik seviyesinde yatmaktadır. Daha verimli ağ kaynakları kullanımı, değişen ağ koşullarına daha hızlı uyum, merkezileştirilmiş kontrol aracılığıyla geliştirilmiş güvenlik ve yeni hizmetlerin veya değişikliklerin daha kolay uygulanması gibi avantajlar sunar. Bu, yüksek ölçeklenebilirlik, çeviklik ve güvenlik talep eden ortamlarda modern ağcılıkta SDN ve bileşenlerinin, örneğin güney yönlü iletişim, merkezi bir rol oynamasını sağlar.

Güney yönlü iletişim, SDN'de kritik bir bileşen olarak, SDN denetleyicisi ile temel ağ altyapısı arasındaki hayati iletişim bağını oluşturur. SDN, ağ yönetiminde yenilikçi bir yaklaşım bağlamında, bu arabirim, dinamik ve verimli ağ yapılandırması ve işletimi sağlamak için hayati öneme sahiptir. Bu, anahtarlar, yönlendiriciler ve diğer temel ağ bileşenleri gibi bir dizi cihazı içerir. Bu arabirimin temel işlevi, SDN denetleyicisinin ağ kaynaklarını talep etmesine ve özel ihtiyaç ve gereksinimlere göre ağın davranışını düzenlemesine yardımcı olmaktır.

Bu arabirimin SDN mimarisindeki önemi abartılamaz. SDN denetleyicisi tarafından belirlenen yüksek seviyeli, soyut politika ve kuralları, ağ cihazlarının anlayabileceği ve üzerine hareket edebileceği somut, uygulanabilir komutlara çevirmedeki bu çeviri ve iletişim yeteneği, SDN'nin tam potansiyelini gerçekleştirmek için hayati öneme sahiptir.

Ancak, güney yönlü iletişimin SDN mimarilerine entegrasyonu, güvenlikle ilgili zorluklar ve riskler olmadan gelmez. Birincil endişelerden biri, güvenlikle ilgili olarak arabirimin kritik bir iletişim kanalı olmasıdır. Bu arabirim üzerinde yetkisiz erişim veya manipülasyon, ciddi ağ kesintilerine, veri ihlallerine ve ağ bütünlüğünün tehlikeye girmesine yol açabilir.

Bir başka endişe, uyumluluk ve standardizasyonla ilgilidir. Ağ cihazlarının çeşitliliği, her birinin farklı iletişim protokolleri ve arabirimleri olabileceği anlamına gelir ve bu da güney arabirimi aracılığıyla sorunsuz ve verimli bir iletişimin sağlanmasında bir meydan okumayı beraberinde getirir. Bu, çeşitli ağ bileşenleri arasında tutarlı ve güvenilir bir iletişim garanti edebilecek standartlaştırılmış protokollerin ve arabirimlerin geliştirilmesini ve benimsenmesini gerektirir.

Ayrıca, güney arabiriminin sağlamlığı ve güvenilirliği hayati öneme sahiptir. Bu arabirimdeki herhangi bir başarısızlık veya verimsizlik, önemli ağ performansı sorunlarına yol açabilir. Kesintisiz ağ operasyonlarını sağlamak için bu arabirimin yüksek dayanıklılık ve hata toleransı ile tasarlanması ve uygulanması esastır.

Bu çalışma, bir OpenFlow Protokolü ve Destek Vektör Makinesi kullanılarak SDN'de paket sınıflandırma için yenilikçi bir çerçeve sunmaktadır. Bu çerçevenin merkezi odağı, SDN ortamlarında güney iletişimi güvenliğini ve verimliliğini artırmaktır, bu da ağların yönetimi ve işletimi için kritik bir bileşendir.

Bu çerçevenin temel yeniliği, SVM'nin OpenFlow iletişim sürecine entegrasyonunda yatmaktadır. SVM, desen tanıma ve sınıflandırma görevlerindeki etkinliğiyle tanınan saygın bir makine öğrenimi modelidir. SDN bağlamında, SVM, ağ güvenliğini ve performansını sürdürmek için hayati olan ağ paketlerini analiz etmek ve sınıflandırmak için kullanılır.

Bu yaklaşımın benzersiz bir yönü, ayarlanmış ağırlık seviyesi metodolojisinin tanıtılmasıdır. Bu yenilikçi teknik, Destek Vektör Makinesi modelini, güney SDN iletişim verilerindeki karmaşık desenleri ve kategorileri daha doğru bir şekilde tanımak için ince ayar yapmayı içerir. OpenFlow protokolündeki farklı parametrelerin ağırlık seviyelerini ayarlayarak, SVM modeli, ağ trafiğinde mevcut olabilecek çeşitli ağ paketleri türleri arasında daha etkili bir şekilde ayrım yapabilir. Bu, ağ trafiğinde mevcut olabilecek potansiyel güvenlik tehditlerini tanımlamak ve hafifletmek için özellikle önemlidir.

Bu makale, SDN'de paket sınıflandırmasına yönelik yenilikçi bir yaklaşım sunuyor ve SVM'nin güçlerini yenilikçi bir ayarlanmış ağırlık seviyesi metodolojisiyle birleştiriyor. Bu çerçeve, SDN ortamlarının güvenlik ve performansını artırmada değerli bir araç olmayı vaat ediyor ve bu ağlarda güneye yönelik iletişim yönetiminin zorluklarına sofistike bir çözüm sunuyor.

Bu araştırmada sunulan çözüm, sınıflandırma modelinin skoruna göre %98,5'lik etkileyici bir tespit oranı ile doğruluk ve güvenilirlik açısından olağanüstü sonuçlar göstermiştir. Bu yüksek doğruluk seviyesi, özellikle SDN ortamlarındaki ağ trafiğinin karmaşık doğası göz önüne alındığında, önemlidir. Ağ paketlerini bu kadar hassas bir şekilde tespit etme ve sınıflandırma yeteneği, ağ güvenliği ve yönetimi alanında önemli bir ilerlemedir.

Bu çerçevenin en dikkate değer başarılarından biri, son derece düşük yanlış alarm oranıdır. Bu yön, ağ yönetiminde özellikle önemlidir, çünkü yanlış alarmlar gereksiz kesintilere yol açabilir ve ağ operasyonlarının genel verimliliğini azaltabilir. Bu yanlış pozitiflerin azaltılması, çerçevenin tasarımında ve uygulanmasında yüksek bir

sofistikasyon seviyesini gösterir ve ağ yöneticileri için güvenilir bir araç olmasını sağlar.

OpenFlow paket sınıflandırma merkezli bu çerçeve, sadece mevcut uygulamasında değil, aynı zamanda gelecekteki araştırmalar için de sağlam bir temel oluşturuyor. Daha gelişmiş güvenlik mekanizmalarının SDN ortamlarında uygulanması için çok yönlü ve etkili bir platform sağlar. Bu çerçevenin uyum sağlama kapasitesi ve etkinliği, özellikle SDN bağlamlarında gelişen güvenlik zorluklarına yönelik olarak daha fazla keşif ve geliştirme için ideal bir aday yapar.

Bununla birlikte, çerçevenin SDN ortamlarındaki yaygın güvenlik risklerini hafifletme potansiyeli son derece umut vericidir. Siber tehditlerin artan karmaşıklığı ve sofistikasyonu göz önüne alındığında, SDN'de sağlam ve uyarlanabilir güvenlik çözümlerine olan ihtiyaç her zamankinden daha kritiktir.

Sonuç olarak, yüksek tespit doğruluğu ve düşük yanlış alarm oranı ile bu çerçeve, bu güvenlik ihtiyaçlarını karşılamada önemli bir adım sunar. Başarısı, SDN güvenliğinde araştırma ve geliştirme için yeni yollar açar ve potansiyel olarak daha güvenli ve verimli ağ ortamlarına yol açar.

1. INTRODUCTION

Software-Defined Networking signifies a revolutionary shift in network architecture, distinguished by the decoupling of the control and data planes. This separation enables a centralized management approach, revolutionizing how network resources are controlled and managed. At the heart of this evolution is the OpenFlow protocol, which is pivotal in enabling this paradigm shift by providing a framework for such decoupling.

SDN's transformative nature lies in its ability to centralize network intelligence in a control plane that is distinct from the data plane, where network devices like switches operate. This is fundamentally different from traditional network architectures, where the control and data planes are intertwined within each network device, leading to complex and rigid networks. SDN, through its centralized control mechanism, promises enhanced network management, agility, and flexibility, allowing for more efficient resource utilization and simplified network configuration and maintenance.

The OpenFlow protocol, a cornerstone of SDN technology, facilitates this separation by defining a standard interface between the control and data planes. It enables the SDN controller, which resides in the control plane, to interact directly with the underlying network hardware. This interaction is crucial for managing the flow of data through the network and for making decisions about how traffic is routed and handled. This southbound communication between the SDN controller and the network devices allows for dynamic, programmable networking, which is a significant departure from the static, hardware-based networking paradigms of the past.

In essence, SDN and OpenFlow collectively represent a paradigmatic leap in networking, introducing levels of programmability, flexibility, and control previously unattainable in traditional network architectures. This paradigm shift not only enables more efficient and flexible network management but also paves the way for innovative network applications and services that can dynamically adapt to changing network conditions and requirements (G. Secinti, 2017).

This interface provides a framework for dynamic allocation of resources, adaptation of behaviors, and instant updates on the network's status. Centralizing such a system brings significant advantages, especially in the realm of advanced security protocol implementation. However, this centralization is not without its own set of security vulnerabilities.

Centralized systems, by their nature, present a unified point of control, which, while beneficial for coordinated management and oversight, also becomes a prime target for security threats. The concentration of control and data in one place means that any breach can have far-reaching and potentially catastrophic consequences.

In the context of dynamic resource allocation, the central system must constantly adjust and reallocate resources based on demands. This agility is a double-edged sword; while it allows for efficient and responsive management of resources, it also requires continuous monitoring and rapid response to any anomalies, which can be challenging to maintain at all times.

Behavioral adjustments in such a system are equally critical. The system must learn and evolve based on usage patterns and potential security threats. This learning process, though crucial for the system's effectiveness and security, also introduces complexity and the need for sophisticated algorithms that can identify and adapt to both regular and irregular patterns.

Network status updates are essential for maintaining the health and security of the system. These updates must be accurate and timely, allowing for immediate action in case of any discrepancies or threats. However, the constant flow of data and the need for its immediate analysis and response demand robust and fail-safe computational capabilities (Mporas, 2021).

Despite these challenges, the centralized approach remains appealing due to its efficiency in managing complex systems and implementing comprehensive security measures. The key lies in balancing the inherent advantages of centralization with the necessary safeguards to mitigate its vulnerabilities. This involves developing advanced security protocols that are robust enough to protect the system while being flexible enough to evolve with emerging threats (Guerroumi, 2019).

To maintain robust network security, it is crucial to implement effective misuse detection and classification systems. This is especially vital in SDN infrastructures,

which are susceptible to a variety of threats and misuse due to their unique architecture. A multi-layered approach is often recommended in this context, as it allows for a more comprehensive defense strategy. (L., 2022)

Advanced technologies play a pivotal role in this strategy. For instance, machine learning algorithms can be employed to analyze network traffic patterns and identify potential threats with greater accuracy than traditional methods. This proactive stance is vital in an era where cyber threats are continuously evolving, becoming more sophisticated and harder to detect.

Moreover, network administrators must remain vigilant and adaptable. This involves not only implementing current best practices but also actively engaging in ongoing research and development. The field of network security is dynamic, with new challenges and solutions emerging regularly. By staying informed about the latest advancements and integrating them into their security protocols, network administrators can better protect SDN infrastructures (T. Han & S. R. U. Jan & Z. Tan & M. Usman & M. A. Jan & R. Khan, 2019). The key to safeguarding SDN infrastructures lies in a multi-faceted approach that incorporates misuse detection, advanced technological tools, and a commitment to continuous learning and adaptation. This approach ensures that networks remain resilient against both current and future cyber threats.

Flow management plays a vital role in securing the southbound communications within SDN's OpenFlow architecture. It encompasses the initiation, monitoring, and conclusion of network traffic flows. For this purpose, two main strategies are employed: statistical methods and classification algorithms. Statistical methods are centered on analyzing traffic patterns, aiming to spot irregularities like unusual surges in data flow or atypical packet sizes. On the other hand, classification algorithms are designed to identify more complex threats. These methods work together to enhance the security and efficiency of network traffic management in SDN environments (Joshi, 2022).

The existing research in SDN using Openflow for packet classification reveals a notable deficiency, particularly in the precise classification of southbound traffic. This challenge arises from the diverse and complex nature of flow control packets, complicating the establishment of uniform classification standards. Current

methodologies often attempt to streamline this process by employing a limited set of features for packet classification (E. Horsanali & Y. Yigit & G. Secinti & A. Karameseoglu & B. Canberk, 2021), inevitably leading to compromised accuracy. Additionally, a major constraint is the laborious and costly acquisition of extensive datasets essential for training, compounded by the necessity of manual labeling for each data point.

1.1 Challenges

This study presents an innovative framework for packet classification in OpenFlow networks, utilizing a SVM model. The core objective of this framework is to maintain high classification accuracy while using a minimal set of features and operating under constraints of limited sample sizes. The choice of an SVM model is strategic; these models are known for their proficiency in managing intricate, multi-dimensional feature spaces. This attribute is particularly beneficial in the context of OpenFlow networks where the data attributes can be extensive and complex.

Moreover, SVM models exhibit exceptional performance in classifying datasets that are not linearly separable, which is a common challenge in network traffic data. The proposed framework leverages these strengths of SVMs, demonstrating their suitability for the specific requirements of OpenFlow packet classification. The research highlights how the integration of SVM models with carefully selected features can lead to efficient and accurate packet classification in OpenFlow environments, which is essential for effective network management and security (Ma, 2008).

1.2 Contribution

This framework introduces three key advancements in network security within SDN. Firstly, it integrates a superior SVM model tailored for OpenFlow packet analysis in SDN environments. This enhancement significantly elevates the detection accuracy of malicious packets by effectively differentiating between legitimate and harmful data traffic.

Secondly, the framework introduces a novel feature weighting mechanism. Features are assigned weights—low, medium, or high—based on their relative importance. This

stratification allows for more refined flow identification and categorization, offering distinctive signatures for each network flow. This method not only aligns with but also augments traditional approaches by incorporating two additional parameters, thereby boosting the precision of packet classification.

Lastly, the framework presents an innovative addition to the SDN architecture: the Flow Listener sub-layer. Positioned within the SDN's southbound interface, this sub-layer plays a crucial role in the enhanced collection and analysis of network traffic parameters. By focusing on detailed traffic data acquisition and evaluation, this sub-layer facilitates a more thorough understanding and management of network flows, thereby strengthening the overall network security posture.

1.3 Organization of the Thesis

The structure of this paper is organized in the following manner:

Chapter 2 presents a comprehensive review of existing research pertinent to the implementation of packet classification in SDN for enhancing security measures. This section delves into various methodologies and approaches documented in the literature, providing a critical analysis of their relevance and applicability in the context of SDN.

In Chapter 3, we introduce our innovative OpenFlow Packet Classification Framework. This section is dedicated to a thorough exposition of the framework's architecture, including a detailed description of its constituent modules and their functionalities. We elucidate how this framework integrates with SDN environments to bolster security protocols.

Chapter 4 is devoted to a rigorous evaluation of our proposed framework. Here, we compare our approach with existing benchmarks, employing a range of training datasets of diverse sizes. This section offers an in-depth analysis of the classification outcomes across various SDN control flows, highlighting the efficacy and efficiency of our framework in different scenarios.

The paper culminates in Chapter 5, where we summarize our findings and discuss potential avenues for future research. This concluding section reflects on the implications of our work for the evolution of SDN and lays out a roadmap for subsequent investigations in this field.

2. RELATED WORK

SDN represents a significant shift in networking by separating the control and data planes. However, SDN's OpenFlow has vulnerabilities, mainly in security (M. Elhejazi & M. Musbah, 2021). These vulnerabilities fall into two categories: implementation-specific and protocol-specific. Implementation-specific vulnerabilities often result from software issues or configuration errors in a specific vendor's version of OpenFlow. In contrast, protocol-specific vulnerabilities stem from OpenFlow's design flaws, like inadequate authorization and authentication mechanisms.

Ahmed et al. (Hafeez, 2020) explored these protocol-specific vulnerabilities in SDN security, examining various policy parameters, such as timeouts, OpenFlow matchfields, and network topologies. They note that if these parameters are easily discoverable in an SDN domain, adversaries can target the SDN data plane. Their solution involves fingerprinting these parameters, tested using the Mininet network emulator (Canberk, 2015). However, their research doesn't cover packet classification, crucial for identifying malicious flows.

Another study (Zappatore, 2017) integrates IDS with SDN, employing SVM for malware detection through traffic features an SDN controller can extract. While this demonstrates SVM's utility in SDN security, it doesn't tailor the SVM model for specific vulnerabilities in OpenFlow packets, a gap our work addresses. Literature on packet classification is extensive but lacks SDN and OpenFlow specificity. Taylor and Turner (Turner, 2007) discuss general packet classification algorithms, not adapted for SDN or OpenFlow. Li et al. (Xie, 2018) use a decision-tree approach for OpenFlow packet classification, achieving good accuracy but with overfitting risks in high-dimensional spaces. Zhang et al. (X. Zhang & G. Xie & X. Wang & P. Zhang & Y. Li & K. Salamatian, 2021) propose a Neural Network model for SDN packet identification, effective for standard protocols but limited for OpenFlow packets due to dataset granularity issues. Recent research (H. Nurwarsito & M. F. Nadhif, 2021) uses Random Forest for SDN packet classification, showing high detection rates but

requiring extensive training data, impractical in data-scarce environments. Prior studies primarily focused on the southbound communication channel, tackling flow table modification and switch spoofing.

Our work introduces a framework for OpenFlow Packet Classification using an enhanced SVM model. We address protocol-specific vulnerabilities with a fine-tuned SVM for OpenFlow packets, achieving higher accuracy in classifying malicious packets. Our model includes an adjusted-weight parameter and a new flow analysis sub-layer in the SDN's southbound interface, improving classification accuracy and laying groundwork for future research.

3. THE PROPOSED OPENFLOW CLASSIFICATION FRAMEWORK

The framework utilizes the SVM model, a key supervised machine learning algorithm, to effectively distinguish between legitimate and malicious OpenFlow messages. As illustrated in Figure 3.1, the core of the OpenFlow packet classification framework is integrated into the Control Layer, with a partial extension into the southbound interface via the Flow Listener module. The orange-dash box in the figure emphasizes the paper's novel contributions to the conventional SDN OpenFlow architecture. Additionally, the flow of data within this framework is depicted using directional arrows. The specific ways in which this paper enhances the traditional OpenFlow SDN architecture are detailed in the following subsections.

3.1 Data Layer

In the SDN architecture, the Data Layer plays a crucial role in efficiently managing packet transmission. Its primary function is to facilitate the movement of packets from the NIC to their designated port on an OpenFlow switch (Liao, 2016). In this setup, the incoming data flows aren't directly processed by the OpenFlow switch. Rather, these flows are mirrored, a strategy that enhances the speed of flow forwarding. This approach allows the Data Layer to focus solely on the rapid transmission of packets, while the responsibility of in-depth flow management is handed over to the SDN Control Layer. To maintain clarity and focus in this explanation, only a single NIC is illustrated in the accompanying Figure 3.1.

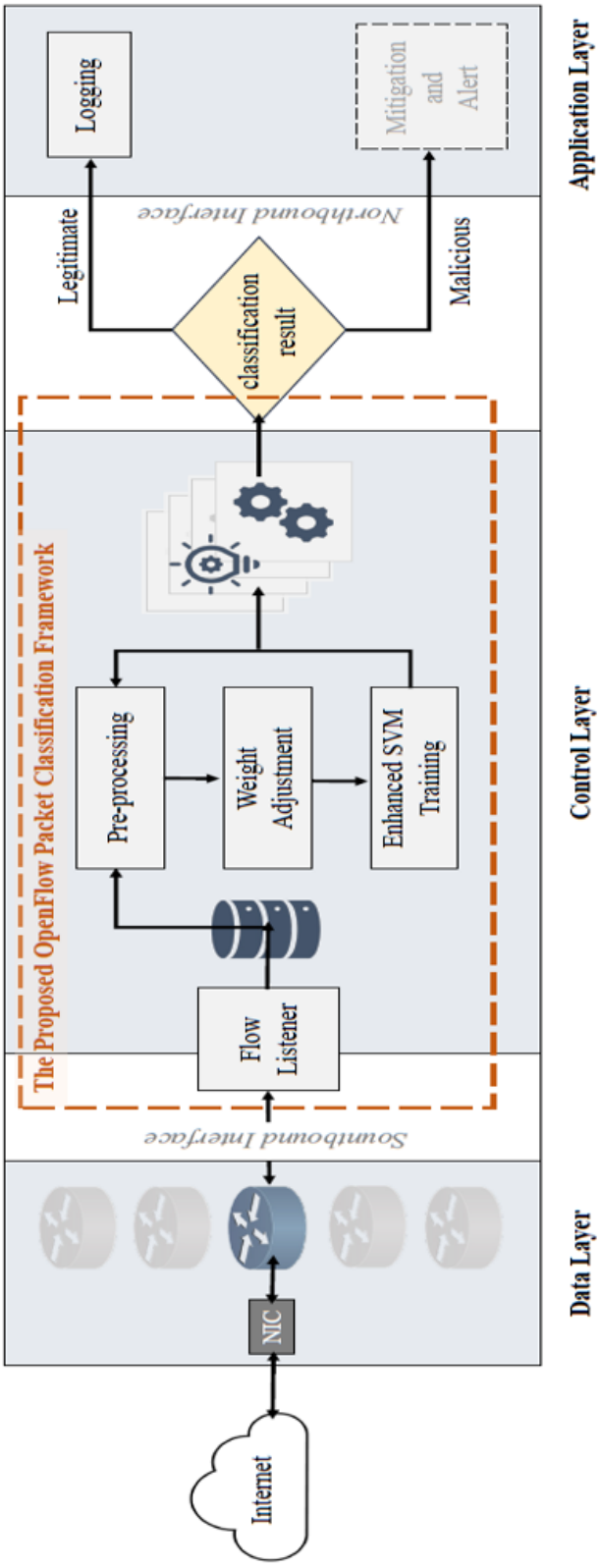

Figure 3.1 : The proposed OpenFlow classification framework.

3.2 Control Layer

The proposed framework, designed for implementation in the Control Layer of SDN, is structured around four primary components: Flow Listener, Preprocessing, Weight Adjustment, and Enhanced Support Vector Machine Training. This framework is meticulously engineered to bolster the security of SDN by adeptly classifying security threats and anomalous behaviors in the southbound communication channel.

3.2.1 Flow listener

The Flow Listener operates as the initial contact point for data acquisition. It systematically collects data from various network devices and routers, laying the groundwork for generating insightful analytics concerning traffic patterns, performance metrics, and potential anomalies. When a new interaction is detected within the SDN flow, its features are meticulously extracted and transferred to a Database, where they undergo subsequent stages of Preprocessing, Weight Adjustment, and Training.

This component focuses on capturing critical features from OpenFlow messages, including the message type, sender IP address, and payload attributes. These features are crucial for the SVM's pattern recognition and data classification capabilities. In this study, we concentrated on eight specific SDN flow control packages: Features, Configuration, Modify-State, Packet Out, Read-State, Role-Request, Barrier, and Asynchronous- Configuration. These flows form the basis of our analysis.

3.2.2 Preprocessing

The Preprocessing step is integral to the data preparation process. It begins with the normalization of data ranges to align data across a uniform scale, thus enhancing the efficiency of the SVM training process (al, 2022).This phase also involves the extraction of vital attributes and characteristics that encapsulate the essence of the network traffic. These attributes are pivotal in defining the nature and intricacies of the network traffic being analyzed.

3.2.3 Weight adjustment

This stage involves the categorization of parameters into three levels of importance: low, medium, and high. Such stratification significantly enhances the effectiveness of SVM training and, by extension, the accuracy of classification (Mishra, 2021).This addresses a notable gap in existing literature concerning the importance of feature-level prioritization in OpenFlow message classification (J. Li & F. Guo & Y. Zhou & W. Yang & D. Ni, 2023). Our SVM studies indicate that feature prioritization during training augments accuracy. However, to our knowledge, no study has classified OpenFlow packages as legitimate or malicious based on eight features with varying importance levels in SDN Southbound Communication. Hence, we have assigned a specific level of importance to each main SDN flow as depicted in Table 3.1. During the parameter determination and weighting process, medium-sized data pools were utilized due to their proven performance efficiency. This stage encompassed an exhaustive exploration of all available parameters and their corresponding weights, systematically evaluating their impact on the intended outcomes.

3.2.4 Enhanced SVM training

In the realm of SDN and OpenFlow vulnerabilities, the SVM plays a pivotal role in classifying network traffic as benign or potentially malicious (Park, 2016). During the training phase, the SVM classifier is utilized with a labeled dataset, enabling it to learn from examples and establish correlations between extracted features and message types. This trained SVM model then becomes the cornerstone for classifying incoming OpenFlow messages as either legitimate or malicious.

3.3 Application Layer

During operation, incoming OpenFlow messages are classified based on the trained SVM model. If any anomalous behavior is detected, the framework initiates appropriate countermeasures, such as alerts or traffic blocking, to safeguard the SDN environment. Conversely, when the classification result is deemed legitimate, the decision is documented for further analysis. In essence, our framework presents a comprehensive OpenFlow packet classification methodology, encompassing the

collection of network traffic flow parameters from a virtual SDN environment, the assignment of adjusted weight levels to features, and the training of the SVM model specifically tailored for OpenFlow packages.

Table 3.1 : The proposed openflow packet's features, levels and descriptions.

Feature	Level	Description
Number of Packets	medium	Represents the total count of packets within a specific flow. It gives an idea of how
Number of Bytes	high	Indicates the total size of data transmitted in the flow. It provides a sense of the magnitude of data transfer between the two communicating entities.
Duration of the Flow in Seconds	medium	Represents the time for which the flow was active. It starts when the first packet of the flow is classified and ends with the last packet.
Byte Rate (Number of Bytes/Duration of the Flow in Seconds)	low	It is the rate at which bytes are being transmitted in the flow, typically measured in bytes per second.
Packet Rate (Number of Packets/Duration of the Flow in Seconds).	low	The rate at which packets are being transmitted in the flow, typically measured in packets per second.
Length of the First Packet	high	Represents the size of the first packet in the flow. It can be important for certain types of analysis, when determining the nature of a flow.
Length of the End Packet	low	Refers to the size of the last packet in the flow. It can be used to classify any anomalies, especially in the context of security.
Inter-arrival Time Between per Packet	high	This denotes the time gap between the arrival of successive packets in a flow.

4. PERFORMANCE EVALUATION

Our evaluation of the effectiveness of our new framework involved a series of experiments conducted on a SDN testbed. We collected a dataset composed of OpenFlow messages from the southbound communication channel of the SDN. These messages were categorized as either legitimate or malicious, providing a reliable basis for training our SVM classifier. The effectiveness of the classifier was evaluated in terms of its ability to accurately classify messages, as well as its rates of False Positives, False Negatives, True Positives , and True Negatives. For comparison, we used a baseline SVM model that does not incorporate our newly proposed adjusted weight method and is trained with six fewer features. This baseline model lacks the two additional features we introduced: the Duration of Flow in Seconds and the Inter-arrival Time Between Packets. These additional features are crucial for enhancing the accuracy and robustness of the classifier in distinguishing between legitimate and malicious messages in the SDN environment.

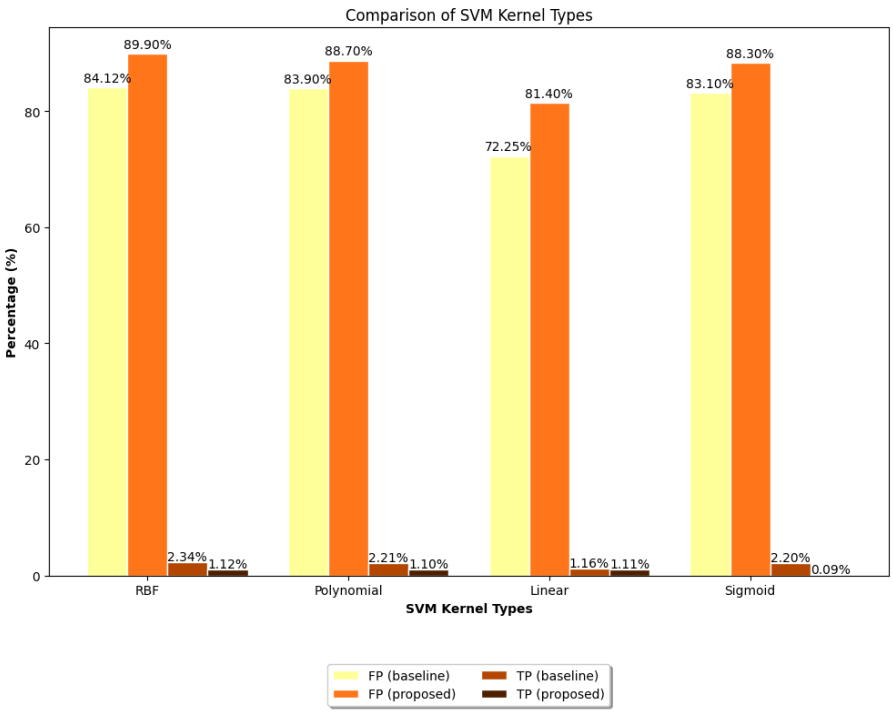

Figure 4.1 : SVM kernel comparison through TP and FP rates.

In our research, we employed a variety of OpenFlow messages, precisely eight distinct types as detailed in Table 4.1. This study primarily focused on the Controller-to-Switch Messages category. A significant aspect of our methodology was the meticulous recording of the distribution of these OpenFlow messages. During the data analysis phase, we achieved a high success rate, with 99.9% of the messages being effectively processed and analyzed. However, it's crucial to acknowledge that a small fraction, specifically 0.1%, of the data encountered issues related to accessibility or retrieval. This minor portion of the data, for reasons not determined, remained beyond our reach for analysis. Consequently, the scope of our transactions and evaluations was inherently limited to the data subset that was fully accessible, which accounts for the vast majority, that is, 99.9% of the messages. This limitation was factored into our study, ensuring that our conclusions and insights are drawn from the comprehensive analysis of the accessible data.

Table 4.1 : OpenFlow message distribution.

No	OpenFlow Message Category	Message Rate
M_1	Features	9,80%
M_2	Configuration	13,50%
M_3	Modify-State	31,70%
M_4	Packet Out	21,70%
M_5	Read-State	12,40%
M_6	Role-Request	4,80%
M_7	Barrier	2,60%
M_8	Asynchronous-Configuration	3,40%

4.1 Testbed Setup and Training

The initial phase of establishing the SDN testbed involved setting up foundational components crucial for a controlled and representative testing environment. Our configuration included integrating the Ryu SDN controller with the Mininet network

emulator, all operating within a CentOS platform. This meticulous arrangement was critical to ensure precision and reliability in our experimental outcomes.

In the realm of our SDN framework, we placed significant emphasis on logging and analyzing classification decisions. This meticulous data collection was instrumental in generating comprehensive results, particularly in constructing a detailed confusion matrix and assessing the overall accuracy of the system. Such an approach was vital for understanding the efficacy of our SDN setup in operational scenarios.

Parallel to the testbed configuration, we embarked on the critical task of determining optimal parameters for the SVM model, specifically tailored for handling OpenFlow SDN packets. The process of selecting and fine-tuning these parameters is a cornerstone in the domain of machine learning, as these variables essentially dictate the functionality and predictive capability of the model. To augment this process, we introduced the concept of adjusted weight feature levels. This innovative approach significantly enhances the parameter selection process by giving more nuanced control over the SDN's flow features and their corresponding levels.

The flow features of the SDN and their associated levels play a vital role in the overall performance of our framework. The judicious selection of these parameters and levels, which we have detailed in Table 3.1, was a pivotal factor in the successful outcome of our experiments. This comprehensive detailing underscores the intricate relationship between the chosen parameters and the operational efficacy of the SDN system.

Additionally, we leveraged the OpenFlow library to its full potential in our experimental endeavors. The library served as a gateway to access and utilize the aforementioned parameters effectively. The use of OpenFlow was not merely a technical choice but a strategic one, ensuring that our experiments were grounded in current industry standards and practices. This alignment with industry standards not only enhances the relevance of our research but also ensures that our findings and methodologies can be directly applied in real-world SDN environments.

The essence of our study hinges on the application of SVM, renowned for their effectiveness in non-linear classification tasks, a topic elaborated upon in Section II. SVM operates by employing kernel functions, essential in mapping data into a higher-

dimensional space, thereby elucidating the separation between various classes. In pursuit of the optimal kernel for OpenFlow packet classification, augmented by our novel adjusted-weight methodology for packet features, extensive experiments were conducted (Cherkaoui, 2013).

Our investigation encompassed four primary SVM kernels: RBF, Polynomial, Linear, and Sigmoid. Selection of the optimal parameters for each kernel was meticulously performed using Grid Search techniques. The evaluation criteria included the TP and FP rates for both our proposed framework and a baseline for comparison. The results, graphically represented in Figure 4.1, underscore the superior performance of the RBF kernel. It stands out with a high TP rate for both the baseline and our framework, coupled with a relatively low FP rate in both contexts. The RBF kernel's overall classification accuracy is notably high, establishing it as a robust choice for our application.

The Polynomial kernel also exhibited strong performance, slightly trailing the RBF kernel in terms of TP rate, yet maintaining a low FP rate across both the baseline and the proposed framework in SDN traffic flow. Its classification accuracy is commendable, although it does not quite match the RBF kernel's efficiency.

Contrastingly, the Linear SVM kernel demonstrated lower TP rates and slightly higher FP rates compared to both the RBF and Polynomial kernels. This resulted in an overall classification accuracy that was inferior to the other kernels examined. The Sigmoid kernel, akin to the Polynomial in TP and FP rates, showed a slight reduction in TP rates and a marginal increase in FP rates compared to the RBF kernel. While its overall classification accuracy is respectable, it does not reach the high standards set by the RBF kernel.

In sum, the RBF kernel emerges as the most proficient among the kernels evaluated. Its high TP rates and low FP rates for both the baseline and our proposed framework make it an exemplary choice for traffic classification in SDN environments. Consequently, in subsequent evaluations, we have employed the RBF kernel with two critical parameters: the Complexity Parameter and the Kernel Parameter.

The Complexity Parameter is set at 5, reflecting our aim to minimize training error. This selection is predicated on the assumption that the training dataset is largely free of label noise, thus being predominantly clean. Meanwhile, the Kernel Parameter is fixed at 2, indicating a decision boundary moderately influenced by training examples. This boundary facilitates a more nimble and effective response during the operational phase of our framework. This study, therefore, provides a comprehensive analysis of SVM kernel performance in the context of SDN traffic flow, underscoring the RBF kernel's suitability for this application.

4.2 Performance Evaluation for Robustness

In this research, extensive experiments were conducted to evaluate the robustness and effectiveness of the proposed OpenFlow classification framework. The primary objective was to ascertain how well the framework performs when trained with varying proportions of a dataset, a critical aspect in determining the practical viability of any classification model. A standard baseline model was employed as a comparative measure to establish a benchmark for the performance of the proposed approach under different training conditions.

The OpenFlow packet classification framework, a cornerstone of this research, was designed to categorize network packets using specific criteria. The novelty of this framework lies in its adjusted-weight tuning mechanism, which was hypothesized to enhance the classification accuracy and efficiency. This research sought to validate this hypothesis by comparing the framework's performance with that of a conventional baseline model across various training scenarios.

Three distinct training scenarios were formulated to assess the adaptability and stability of the proposed framework and the baseline model:

Training with 40% of the Data (Training %40): In this scenario, only 40% of the available data was used to train the classification model. The remaining 60% of the dataset was reserved for testing and validation purposes. This scenario simulates a common real-world condition where limited data is available for training, challenging the model's ability to learn effectively from a smaller sample size.

Training with 60% of the Data (Training %60): Here, a larger portion of the dataset, specifically 60%, was allocated for the training phase. The remaining 40% was used for testing the model. This scenario represents a more balanced approach, providing a substantial amount of data for training while still retaining a significant portion for validation and testing.

Training with 80% of the Data (Training %80): This scenario involved utilizing 80% of the dataset for training the model, leaving only 20% for testing and validation. This setup tests the model's performance when a substantial majority of the available data is used for training, a scenario that could potentially lead to overfitting or high variance in less robust models.

The experimental results, as detailed in Table 4.2, underscore the superior performance of the proposed OpenFlow classification framework, particularly when compared to the baseline model. The data clearly demonstrates that the proposed framework, highlighted in blue text in the table, exhibits less fluctuation in its performance across different training scenarios. Specifically, the framework showed only a 0.1% difference in the TP rate and a 0.3% difference in the FP rate when compared to the baseline model across these scenarios. This indicates a higher level of stability and reliability in the framework, even with lesser amounts of training data.

These findings are significant as they illustrate not only the effectiveness of the proposed OpenFlow packet classification framework but also its robustness and adaptability to different training conditions. The adjusted-weight tuning component of the framework appears to contribute positively to its overall performance, making it a promising tool for efficient and accurate packet classification in various network environments. This research thus provides valuable insights into the development of more effective classification models in the field of network management and security.

Table 4.2 : Three different trainin scenarios.

Dataset	TP Rate	FP Rate
Training %40, Baseline	0.8386	0.0244
Training %40, Our Framework	0.8972	0.0133
Training %60, Baseline	0.8410	0.0239
Training %60, Our Framework	0.8980	0.0121
Training %80, Baseline	0.8412	0.0234
Training %80, Our Framework	0.8990	0.0112

4.3 Performance Evaluation for Classification

In this study, we implemented an 80% training/test split methodology using a RBF kernel for SVM to assess the efficacy of our classification model. The performance metrics were evaluated under two scenarios: one without our proposed framework (baseline implementation) and one with it.

The baseline model's performance, as depicted in Figure 4.2(a), demonstrated a True Positive rate of 84.12% in correctly identifying “Legitimate” internet traffic. However, it also exhibited a False Negative rate of 6.12%, indicating a tendency to misclassify some genuine traffic as “Malicious.” The False Positive rate and True Negative rate were recorded at 2.34% and 7.42% respectively. These statistics highlight the model's general effectiveness but also underscore the need for enhanced precision in distinguishing malicious traffic.

In contrast, the introduction of our proposed framework led to a notable improvement in classification accuracy, as seen in Figure 4.2(b). The TP rate increased significantly to 89.9%, and the FN rate decreased markedly to 0.34%. This substantial reduction in FN rate underscores the framework’s heightened accuracy in classifying traffic.

Moreover, the FP and TN rates were observed at lower levels of 1.12% and 8.64%, respectively, further endorsing the enhanced capability of the proposed framework in accurately segregating “Legitimate” and “Malicious” traffic.

The overall performance of our proposed model was quantitatively superior to the baseline model and traditional SVM approaches, as indicated by the following metrics: an Accuracy of 98.54%, a Precision of 97.77%, a Recall of 99.62%, and an F1-score of 99.19%. These values not only reflect the model's high efficiency in traffic classification but also emphasize the significant advancements our method offers over conventional techniques.

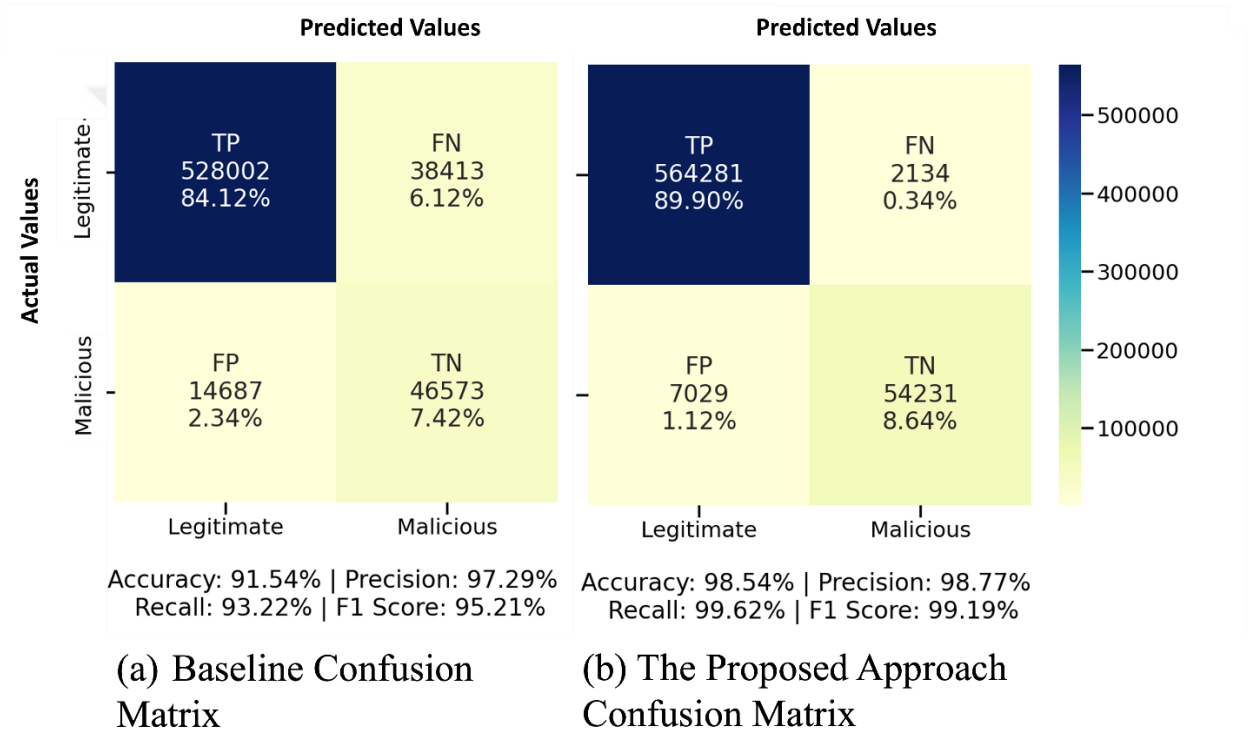

Figure 4.2 : Confusion matrix results.

The study conducted a comprehensive evaluation of the classification outcomes in SDN using OpenFlow packet categories. This assessment aimed to determine which SDN flows are more prone to being erroneously classified as malicious under the proposed SVM-based OpenFlow classification framework. The results, illustrated in Figure 4.3, revealed that Modify-State (M_3) flows experienced higher FP rates. In simpler terms, a significant number of Modify-State flows were incorrectly labeled as “malicious”.

This finding is particularly noteworthy given the nature of SDN architecture. Modify-State flows, which are primarily responsible for adding, deleting, or modifying flow table entries and setting switch port priorities, are predominantly communicated through the Southbound Interface to the OpenFlow switches. In contrast, Barrier (M_7) flows, which serve different purposes in the network, were found to be the least misclassified among the eight evaluated SDN flow categories. This disparity in classification accuracy can be attributed to the inherent differences in the frequency and nature of these flows within the SDN architecture, as further corroborated by the OpenFlow distribution data presented in Table 4.1.

The study's experimental setup showcased the effectiveness of the SVM-based classification framework in accurately distinguishing between legitimate and malicious messages within the southbound communication channel of an SDN environment. The framework's proficiency was evident in its low rates of both false positives and false negatives. This level of accuracy is critical in SDN contexts, where the precise identification of traffic nature is paramount for network security and efficiency.

Overall, the research underscores the viability of using an SVM-based approach for the classification of OpenFlow messages in SDN environments. The framework not only demonstrates high accuracy but also ensures a reliable identification process, which is essential for maintaining the integrity and performance of SDN networks. This study, therefore, contributes valuable insights into the development of more secure and efficient SDN systems by highlighting the importance of tailored classification strategies for different types of SDN flows.

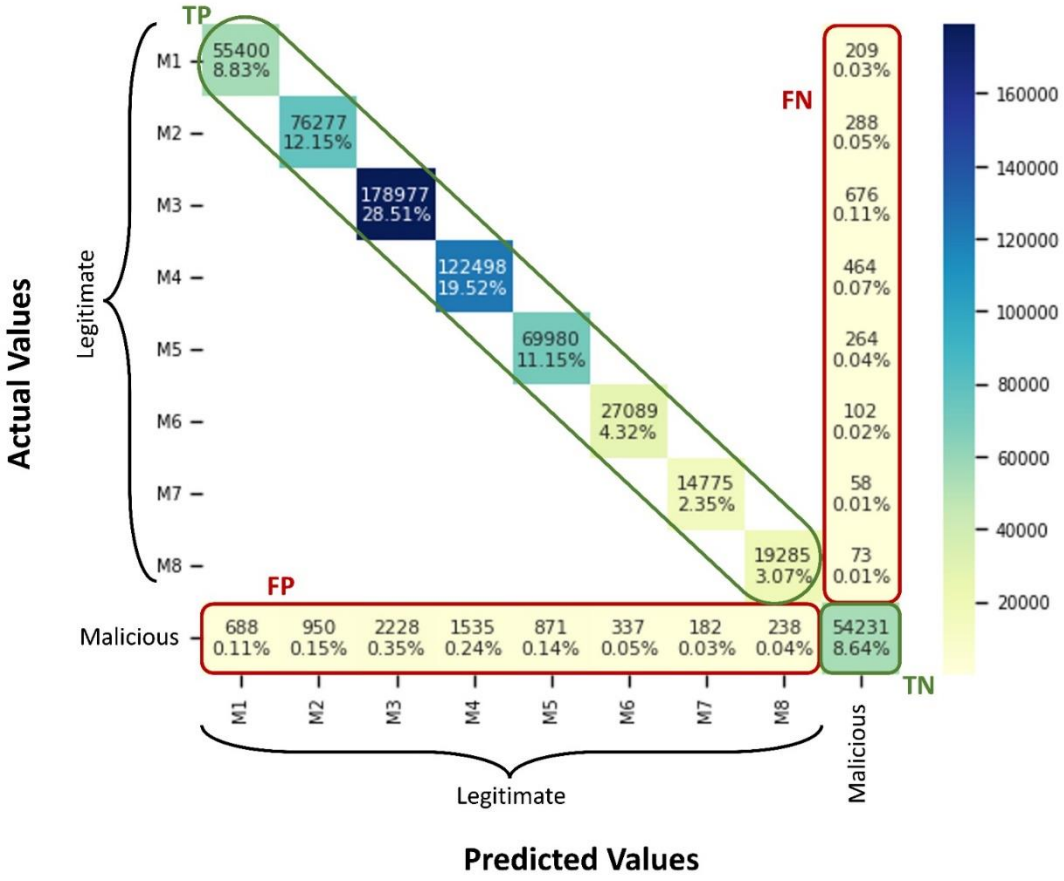

Figure 4.3 : The detailed confusion matrix for the proposed approach.

5. CONCLUSIONS AND RECOMMENDATIONS

This study introduces an innovative framework for classifying OpenFlow messages in the southbound communication channel of SDN using SVM algorithms. The integration of machine learning techniques into this framework plays a crucial role in accurately identifying OpenFlow communications. This capability is fundamental for detecting potential security threats in the SDN's control layer.

A distinctive aspect of our approach is the use of specific level constants for parameter configuration. This strategic selection has significantly improved the framework's ability to pinpoint anomalies, setting it apart from traditional classification methods. The conducted experiments affirm the effectiveness of our methodology, showcasing its advantages over existing techniques.

Looking ahead, there are several avenues for further enhancement of this framework. A primary focus would be on improving its scalability, ensuring it can handle increasingly large and complex network environments. Additionally, adapting the framework to recognize and mitigate newly emerging attack vectors remains a critical area of development.

Another potential improvement involves refining the framework's efficiency. This can be achieved by adjusting the levels of parameters used in the classification process and expanding the data set utilized for defining these parameters. Such modifications are anticipated to yield more precise and efficient outcomes in identifying and classifying security threats in SDN environments. This paper, therefore, lays the groundwork for more advanced and robust SDN security measures, opening pathways for future research and development in this fields.

REFERENCES

al, Z. S. (2022). Block Error Rate Analysis of Short-Packet Mobile-to-Mobile Communications Over Correlated Cascaded Fading Channels. (pp. 4087-4101). IEEE Transactions on Vehicular Technology.

Canberk, M. E. (2015). Scalability analysis and flow admission control in mininet-based sdn environment. (pp. 18-19). IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN).

Cherkaoui, T. S. (2013). Optimal packet classification applicable tothe OpenFlow context. (pp. 9–14). Association for Computing Machinery.

E. Horsanali & Y. Yigit & G. Secinti & A. Karameseoglu & B. Canberk. (2021). Network-aware automl framework for software-defined sensor network. (pp. 451-457). 17th International Conference on Distributed Computing in Sensor Systems (DCOSS).

G. Secinti, P. B. (2017). Resilient end-to-end connectivity for software defined unmanned aerial vehicular networks. IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).

Guerroumi, S. B. (2019). Intrusion detection system for sdn network using deep learning. Int. Conf. on Theor. and Applicative Aspects of Comput. Sci. (ICTAACS).

H. Nurwarsito & M. F. Nadhif. (2021). Ddos attack early detection and mitigation system on sdn using random forest algorithm and ryu framework. (pp. 178-183). 8th International Conference on Computer and Communication.

Hafeez, B. A. (2020). Fingerprinting sdn policy parameters: An empirical study. (pp. 142379–142392). IEEE Access.

J. Li & F. Guo & Y. Zhou & W. Yang & D. Ni. (2023). Predicting the severity of traffic accidents on mountain freeways with dynamic traffic and weather data. Transportation Safety and Environment.

Joshi, A. G. (2022). Multilayer statistical intrusion detection model for wireless network., (pp. 1-7).

L., Y. H. (2022). Network security situation assessment with network attack behavior classification. *Int J Intell Syst.*, 37: 6909-6927.

Liao, J. -C.-T.-C. (2016). Design the DNS-Like Smart Switch for Heterogeneous Network Base on SDN Architecture. *2016 International Computer Symposium (ICS)*, 187-191.

M. Elhejazi & M. Musbah. (2021). Dynamic defense in-depth model for SDN control layer to enhance OpenFlow protocol security. The 7th International Conference on Engineering (MIS).

Ma, R. Z. (2008). An improved SVM method p-SVM for classification of remotely sensed data. *International Journal of Remote Sensing*, 6029-6036.

Mishra, S. K. (2021). Scalable kernel-based SVM classification algorithm on imbalance air quality data for proficient healthcare. (pp. 2597-2615). Complex Intelligent Systems.

Mporas, M. R. (2021). Privacy and Security Threats from Smart Meters Technology. 2021 International Carnahan Conference on Security Technology (ICCST).

Park, T. V. (2016). A Novel Hybrid Flow-Based Handler with DDoS Attacks in Software-Defined Networking. (pp. 350-357). Toulouse, France: Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld).

T. Han & S. R. U. Jan & Z. Tan & M. Usman & M. A. Jan & R. Khan, a. (2019). A comprehensive survey of security threats and their mitigation techniques for next-generation SDN controllers. *Concurrency and Computation: Practice and Experience*, 32:16.

Turner, D. E. (2007). Classbench: A packet classification benchmark. *IEEE/ACM Trans. on Netw.*, 15:3.

X. Zhang & G. Xie & X. Wang & P. Zhang & Y. Li & K. Salamatian. (2021). Fast online packet classification with convolutional neural network. *IEEE/ACM Transactions on Networking*.

Xie, W. L. (2018). Cutsplit: A decision-tree combining cutting and splitting for scalable packet classification. IEEE INFOCOM 2018 - IEEE Conference on Computer Communications.

Zappatore, L. B. (2017). Support vector machine meets software defined networking in ids domain. (pp. 25-30). 29th International Teletraffic Congress (ITC 29).

CURRICULUM VITAE

Name Surname **:** Ali Gökhan Avran

EDUCATION **:**

- **B.Sc.** **:** 2017, Anadolu University, Engineering Faculty, Electrical Electronic Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2022-Present Senior Software Engineer at Argela Technologies
- 2020-2022 Experienced Software Engineer at AirTiesWirelessNetworks
- 2018-2020 Embedded Software Engineer at Baykar Technologies

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Ali Gökhan Avran**, Elif Ak, Kübra Duran , Gökhan Yurdakul, Gökhan Seçinti**,** Securing Southbound Interface in SDNs: Utilizing Support Vector Machines for OpenFlow Packet Classification, 2023 IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (IEEE CAMAD 2023)